



## SISTEMA DE MÀRQUETING PER PROXIMITAT USANT BLUETOOTH

Memòria del projecte de final de carrera corresponent  
als estudis d'Enginyeria Superior en Informàtica pre-  
sentat per Samuel Jiménez Díaz i dirigit per Ramon  
Martí i Antonio Jiménez.

Bellaterra, Juny de 2010

El firmant, Ramon Martí Escalé, professor del Departament d'Enginyeria de la Informació i de les Comunicacions de la Universitat Autònoma de Barcelona

CERTIFICA:

Que la present memòria ha sigut realitzada sota la seva direcció per Samuel Jiménez Díaz

Bellaterra, Juny de 2010

---

Firmat: Ramon Martí Escalé

*A la meva família, pel seu recolzament en tot moment, on  
sigui i com sigui*



# Agraïments

Gràcies a tots els companys que m'han acompanyat en aquest viatge per la universitat durant els 5 anys de travessia que ha durat.

Gràcies als professors, per ensenyar-me moltes més coses fora de les matemàtiques.

Gràcies a tots.



# Índex

<b>1</b>	<b>Introducció</b>	<b>1</b>
1.1	L'empresa . . . . .	1
1.2	La raó de ser del producte . . . . .	2
1.3	Objectius . . . . .	4
1.4	Estructura de la memòria . . . . .	5
<b>2</b>	<b>Viabilitat i planificació</b>	<b>7</b>
2.1	Viabilitat operativa . . . . .	7
2.2	Viabilitat tècnica . . . . .	7
2.3	Viabilitat legal . . . . .	8
2.4	Conclusions viabilitat . . . . .	8
2.5	Planificació temporal del treball . . . . .	8
2.5.1	Explicació de les tasques . . . . .	9
2.6	Llicències . . . . .	10
<b>3</b>	<b>Especificacions</b>	<b>13</b>
<b>4</b>	<b>Estat de l'art</b>	<b>17</b>
4.1	Entorn . . . . .	17
4.1.1	Hardware . . . . .	17
4.1.2	Sistema operatiu . . . . .	17
4.1.3	Base de dades . . . . .	18
4.2	Bluetooth . . . . .	18
4.2.1	Breu repàs a la història del Bluetooth . . . . .	18

4.2.2	Detalls tècnics . . . . .	19
4.2.3	Més dades tècniques . . . . .	19
4.2.4	Bluez . . . . .	23
<b>5</b>	<b>Disseny i implementació</b>	<b>25</b>
5.1	Diagrama del sistema . . . . .	25
5.2	Llibreries necessàries . . . . .	26
5.3	Base de Dades . . . . .	27
5.3.1	Requeriments de dades . . . . .	27
5.3.2	Disseny lògic . . . . .	28
5.4	Fase 1 - Cerca . . . . .	30
5.5	Fase 2 - Exploració . . . . .	35
5.6	Fase 3 - Enviament . . . . .	38
5.7	Interfície . . . . .	41
5.7.1	La pàgina d'identificació . . . . .	42
5.7.2	El resum general . . . . .	42
5.7.3	Configuració . . . . .	44
5.7.4	Estadístiques . . . . .	45
5.7.5	Llista negra . . . . .	45
5.7.6	Canviar paraula de pas . . . . .	46
5.7.7	Rutines de control . . . . .	46
5.8	Problemes . . . . .	47
<b>6</b>	<b>Proves i resultats</b>	<b>49</b>
6.1	Conclusions . . . . .	50
<b>7</b>	<b>Rendibilitat</b>	<b>53</b>
7.1	Compra del sistema . . . . .	53
7.2	Lloguer del sistema . . . . .	54
7.3	Conclusions . . . . .	55
<b>8</b>	<b>Conclusions</b>	<b>57</b>
8.1	Millores . . . . .	58



8.2 Desenvolupament temporal . . . . .	59
<b>Bibliografia</b>	<b>61</b>



# Índex de figures

1.1	Piràmide de Maslow . . . . .	3
2.1	Diagrama de Gant . . . . .	8
3.1	Hardware on s'instal·larà el sistema . . . . .	15
4.1	Xarxa - piconets . . . . .	20
4.2	Protocols Bluetooth . . . . .	22
5.1	Diagrama de flux del sistema . . . . .	26
5.2	Disseny lògic de la base de dades . . . . .	29
5.3	Cerca de dispositius usant mètode per defecte . . . . .	32
5.4	Cerca de dispositius versió millorada . . . . .	35
5.5	Base de dades després d'aplicar una cerca de dispositius . . . . .	36
5.6	Resultat per pantalla d'aplicar la fase 2 . . . . .	37
5.7	Estat de la Base de dades després d'aplicar una exploració als dispositius localitzats . . . . .	39
5.8	Sortida per pantalla de la fase 3 . . . . .	41
5.9	Estat final després de tots els processos . . . . .	42
5.10	Pàgina d'identificació . . . . .	43
5.11	Pantalla inicial . . . . .	44
5.12	Pantalla de configuració . . . . .	45
5.13	Pantalla d'estadístiques . . . . .	46
6.1	Gràfica punt de saturació del throughput . . . . .	52

8.1	Desenvolupament temporal final . . . . .	60
-----	--	----

# Capítol 1

## Introducció

A continuació es farà una breu introducció al treball. Aquest projecte ha sigut desenvolupat sota la tutela de l'empresa Tecnotor, S.L. La introducció explicarà les motivacions de l'empresa, els seus objectius en aquest projecte i com s'estructurarà aquesta memòria per tal de conèixer el fil conductor.

### 1.1 L'empresa

Tecno-tor S.L. és una empresa fundada l'any 1981, inicialment dedicada a la mecanització de segones operacions en peces d'automoció. Fa uns 15 anys va decidir enfocar la seva producció al decoletatge sent, en aquests moments, la seva principal activitat. Fa prop d'un any, va absorbir una empresa dedicada al control numèric ampliant el seu públic objectiu<sup>1</sup>. Actualment, està en procés creixement buscant nous sectors de mercat per expandir-se ja que el sector automobilístics ha sigut dels més perjudicats per la crisi econòmica actual i han decidit dur a terme una estratègia de diversificació empresarial. S'ha creat un departament d'informàtica (anomenat Nhashi) que està dedicat a crear una aplicació que serveixi per maximitzar la gestió de capital humà a partir del control dels treballadors (mitjançant TAGs RFID, càmeres...).

---

<sup>1</sup>El públic objectiu són les persones a les quals van dirigides les accions o estratègies de l'empresa

Gràcies als seus contactes, se li va presentar la oportunitat de dissenyar un producte de publicitat per proximitat. Varies empreses tals com restaurants o locals nocturns li havien suggerit la possibilitat d'aprofitar el nou personal per fer una aplicació que enviés publicitat degut a què hi havia un alt cost en promocions (imprimir papers i cartells que acabaven al terra trepitjats). Així, és necessari un aparell que sigui capaç de gestionar les comunicacions Bluetooth de l'entorn per actuar en conseqüència i poder enviar missatges de publicitat.

Després d'estudiar-ho, es va decidir prosseguir amb els dos projectes: l'anterior, i l'actual que correspondria a aquest sistema que enviarà publicitat automàticament.

## 1.2 La raó de ser del producte

Actualment, no és suficient en només vendre un producte. Es necessiten estratègies més eficients i profundes per tal d'assolir l'èxit empresarial. Per entendre aquest concepte, hem de veure l'evolució de les estratègies de màrqueting desenvolupades des del inici del *boom* empresarial:

Fa un segle, la demanda era molt superior a l'oferta, de manera que l'estratègia a seguir era produir el màxim d'unitats. Com més unitats, més vendes s'aconseguien, ja que el producte no era tan important degut a què la gent comprava i comprava (al voltant de l'any 1920).

Amb el temps això va canviar. Ja no hi havia tantíssima demanda i sí diferents competidors a tenir en compte pel domini del mercat. Es suposava que si el client tenia moltes opcions a escollir, la seva decisió es basaria en maximitzar la qualitat del producte. En aquell temps, sobre els anys 60 i 70, la demanda era bastant equivalent a la oferta.

Finalment, el creixement econòmic ha permès un mercat amb una altíssima competència (i totalment imperfecte ja que el preu és molt distant al cost marginal<sup>2</sup>) i ha arribat un punt en el que la oferta és molt superior a la deman-

---

<sup>2</sup>La competència perfecta es dona quan les empreses només cobreixen els costos de producció (no hi ha beneficis extra) i per tant, els clients maximitzen la seva compra i l'empresa no surt perdent. A la pràctica, aquesta situació és irreal i l'empresa sempre té beneficis afegits.

da. Per tant, ja no és suficient amb produir moltes unitats o el millor producte del món ja que la segmentació del mercat és molt elevada i la societat necessita quelcom més. *¿Què més?*

Primer, hem de saber en què es fonamentarà la resposta. Abraham Maslow va definir (1943) que els humans tenim una sèrie de necessitats a resoldre. Inicialment, les més bàsiques: alimentació, respirar... Necessitats fisiològiques. Un cop han estat satisfetes (les més importants) es passa al següent nivell (creant una piràmide de necessitats com es mostra a la figura 1.1) on està situada la seguretat. Ens volem sentir segurs. A mesura que anem escalant nivells, sempre n'apareixeran de nous i, per definició, mai serem satisfets.

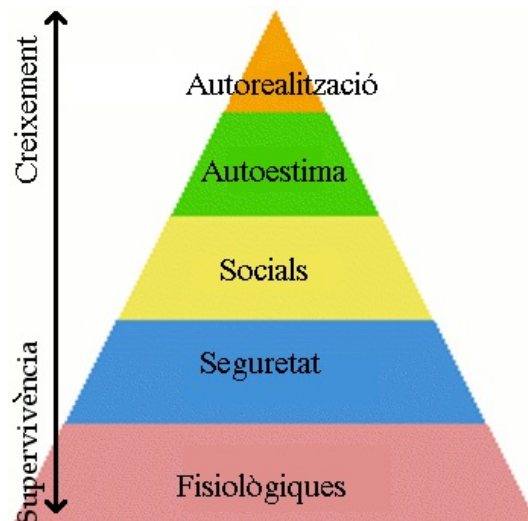


Figura 1.1: Piràmide de Maslow

Gràcies a Maslow, els empresaris es van adonar que el producte només era una part d'un conjunt molt més gran. Quan una persona va a un restaurant, a més a més de menjar bé, també busca un entorn saludable (per exemple, el Hard Rock pels joves o bé l'Starbucks per a la classe mitjana-alta). I aquí és on el nostre producte entra en acció: l'anunciant haurà de dissenyar una imatge (que sigui animada pot ser una molt bona idea) que pugui transmetre els valors que vol vendre, la identitat de l'empresa. Modernitat, serietat, baixos costos (enviant ofertes), etc.

El projecte es basa en dissenyar un sistema que enviarà un missatge de pub-

licitat a les persones que estiguin al voltant del lloc on s'instal·li. Serà feina del consumidor<sup>3</sup> dissenyar una bona imatge per publicitar-se: la solució final serà el medi pel qual farà arribar la seva publicitat.

### 1.3 Objectius

Fins ara, hem parlat bastant de màrqueting i la raó de ser del sistema. Però, *què busquem?* El consumidor no ha de saber res d'informàtica i, per tant, hem de suposar que el seu coneixement serà sempre nul en aquest camp. Conseqüentment, hem d'aconseguir un entorn senzill i intuïtiu.

El sistema serà extern ja que haurà de situar-se el més aprop del públic objectiu possible degut a l'abast limitat dels dispositius Bluetooth. Conseqüentment, haurem d'accedir-hi remotament per tal de configurar-lo ja que seria totalment incòmode haver de muntar i desmuntar contínuament. Partint d'aquesta base, serà necessari que no hi hagi problemes depenen de la plataforma des de la qual accedim. Per això, s'ha de buscar la compatibilitat en qualsevol cas.

Per últim, és molt important tenir present en quin entorn s'estarà movent el sistema: serà sempre variant. La gent passarà caminant a certa velocitat pel radi d'acció pel sistema, per tant, la seva existència contínua dins dels límits no és segura. Això implicarà que s'ha d'arribar a enviar la publicitat tan ràpid com sigui possible. Per aquest propòsit, s'haurà de fer un disseny dels programes concurrents per poder aprofitar tan com sigui possible varies operacions a la vegada.

Resumint, el conjunt d'objectius a aconseguir en el projecte és el següent:

- Aprofundiment en la tecnologia Bluetooth per conèixer les seves possibilitats i fronteres
- Disseny d'una interfície gràfica minimalista i intuitiva, sense problemes de compatibilitat

---

<sup>3</sup>El consumidor és el comprador ja que si bé no rebrà la publicitat, és ell qui l'usarà i comprarà el producte



- Disseny d'una base de dades amb la informació necessària per gestionar les comunicacions Bluetooth que pugui dur a terme el sistema
- Detecció, exploració i interacció amb dispositius mòbils Bluetooth dins del radi d'acció del sistema global
- Concurrencia de processos per evitar la limitació del sistema, aconseguint que aquest pugui tenir un ús professional
- Realitzar un testeig per comprovar el correcte funcionament del sistema
- Exposició i defensa del projecte

## 1.4 Estructura de la memòria

Aquest ha sigut el primer capítol. La resta de la memòria està estructurada tal com s'explica a continuació:

**Capítol 2** En aquest capítol veurem l'estat de l'art del projecte. Quin entorn utilitzarem i perquè, especificacions de la tecnologia Bluetooth, les bases de dades que s'usaran i els llenguatges de programació escollits juntament amb les raons del per què de tot.

**Capítol 3** Continuarem amb el disseny i la implementació. Com s'ha prosseguit a implementar el sistema i les característiques de cada part del projecte, així com els problemes trobats en el desenvolupament del projecte.

**Capítol 4** Parlarem de les llicències dels programes externs utilitzats per evitar complicacions legals.

**Capítol 5** En aquest capítol veurem les proves i resultats. El sistema és exposat a un cas real en un entorn control·lat i es pot apreciar com va agafant la seva maduresa fins arribar al moment d'enviar la publicitat.

**Capítol 6** Veurem un petit estudi de la rendibilitat del projecte a curt i mig termini.

**Capítol 7** Conclusions finals un cop finalitzat el projecte i possibles millores.



# Capítol 2

## Viabilitat i planificació

### 2.1 Viabilitat operativa

El sistema és accessible des de qualsevol computador sense cap tipus de limitació gràcies al servidor web que s'incorporarà al servidor. Només es requereix d'una connexió en xarxa per accedir-hi. Així, el sistema no té cap inconvenient en quan a possibles accessos al dispositiu tot i no estar connectat físicament a ell.

### 2.2 Viabilitat tècnica

Els protocols implementats en l'actualitat estan programats en C. Les possibles limitacions dependran de la cobertura del dispositiu bluetooth connectat al sistema i dels elements físics (parets...) que hi hagin entre el servidor i l'usuari final. El bluetooth té un rang mig d'abast de 25m, de manera que el sistema està pensat per ser col·locat en llocs on els clients no passin gaire lluny del dispositiu per a què sigui viable i factible enviar la publicitat desitjada. Existeixen alternatives per a millorar el rang de cobertura del bluetooth (hi ha varies classes de dispositius) però no es contemplen perquè l'essència del projecte és la proximitat. Tot i així, serà extensible a aquestes alternatives produint un augment de la distància màxima entre dispositiu i client, per a possibles casos extrems.

## 2.3 Viabilitat legal

En el següent capítol, podrem veure les llicències a considerar en el sistema. No n'hi ha cap totalment restrictiva (totes es poden aconseguir pagant o gratuïtes) i, per tant, considerem que és viable.

## 2.4 Conclusions viabilitat

El projecte no presenta, a priori, cap tipus de problema en la seva viabilitat. Com hem dit, pot ser necessari la compra de llicències en la comercialització del sistema. També és possible que s'hagi d'alliberar alguna part del codi implementat per a complir els requisits de les llicències, tot i que dependrà de les possibles extensions que es facin a posteriori.

## 2.5 Planificació temporal del treball

S'ha fet un diagrama de Gant per tal de poder planificar una correcta estratègia temporal. Podem veure-ho a la figura 2.1.

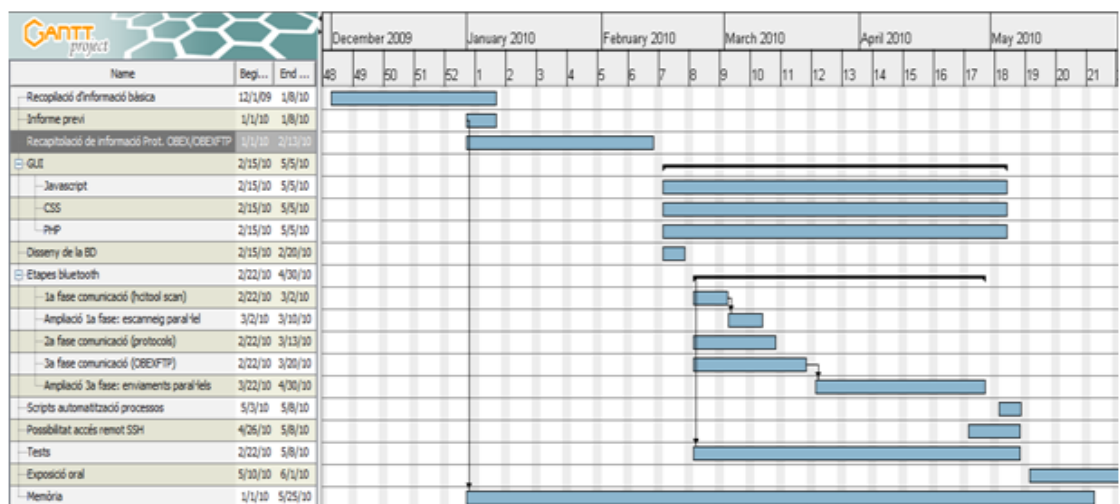


Figura 2.1: Diagrama de Gant

### 2.5.1 Explicació de les tasques

Recopilació d'informació bàsica: informació sobre el projecte a desenvolupar per veure la seva viabilitat, punts forts e interessants. Previ a començar a fer qualsevol part del projecte.

Informe previ: document obligatori pel inici del projecte conforme s'ha estudiat la seva viabilitat i s'ha elaborat una planificació d'aquest.

Recopilació de informació prot. Obex: informació essencial pel desenvolupament de la comunicació amb bluetooth.

GUI: serà desenvolupada al llarg de tot el projecte a mesura que s'incorporen noves característiques al sistema.

1. Codi Javascript: proporcionarà la capacitat d'executar events per part de l'usuari.
2. CSS: permetrà estilitzar la web del panell de control.
3. PHP: gestionarà les accions escollides per l'usuari i permetrà la comunicació entre el servidor i l'administrador. Gràcies al php es podran canviar configuracions, veure resums, etc.

Disseny BD: dissenyar i implementar la base de dades que ens servirà per guardar tota la informació referent a la vida del sistema.

Etapas bluetooth: codi principal del sistema. Tot i ser dependent, la implementació no ho és, així que s'anirà realitzant paral·lelament.

1. 1a fase comunicació (cerca de dispositius): programació de l'aplicació que ens servirà per descobrir dispositius bluetooth al voltant del sistema (possibles *targets*).
2. Ampliació 1a fase (cerca paral·lela): en cas de ser possible, s'implementarà una millora a la 1a fase que permetrà buscar des de diversos *dongles* possibles *targets* més ràpidament aprofitant l'ús paral·lel de *dongles*.
3. 2a fase comunicació (exploració): s'haurà de realitzar un pas intermedi per obtenir informació sobre el target per comprovar que està preparat per rebre

la informació que volem enviar (dependrà del dispositiu de l'usuari, si ve preparat o no).

4. 3a fase comunicació (enviament de fitxers): En aquest pas, un cop s'ha fet el filtre dels dispositius que estan aprop i estan tècnicament preparats pel nostre propòsit, s'implementarà un codi que aprofitant les dades anteriors, enviarà la nostre publicitat.
5. Ampliació 3a fase(enviaments paral·lels): en cas de ser possible, s'implementarà l'enviament paral·lel de publicitat desde diversos *dongles*.

Scripts automatització processos: permetran el manteniment del sistema en tot moment realitzant còpies de seguretat quan sigui oportú, activant/desactivant els serveis, etc.

Possibilitat d'accés remot SSH: s'incorporarà un servidor SSH que permetrà accedir al sistema usant Internet.

Tests: es faran totes les proves pertinents necessàries per assegurar la fiabilitat del producte un cop finalitzat. Tot i així, també hi hauran proves durant el desenvolupament del codi.

Exposició oral: es mostrarà i defensarà el projecte al final d'aquest davant d'un tribunal.

Memòria: es realitzarà durant tot el projecte a mesura que s'avança.

## 2.6 Llicències

Degut a la utilització del sistema operatiu Ubuntu en el sistema, existeixen diverses llicències a considerar en cas de voler utilitzar els seus codis. Alguns són open source, altres software lliure i, en algú cas, s'ha escollit una opció o altre segons si tenia copyright o no el programa.

Passem a veure les diferents llicències que s'han de tenir presents:

- MySQL: té una llicència GNU GPL de forma estàndard, però en cas de voler utilitzar-lo de forma privada s'haurà de comprar una llicència comercial.

- Bluez: té una llicència GNU GPL.
- Ussp-push: té una llicència GNU GPL.
- JpGraph: com MySQL, té una doble llicència. Per raons no comercials, open-source o ús educacional s'utilitza la llicència QPL 1.0 i per raons comercials segueix la llicència JpGraph Professional.





# Capítol 3

## Especificacions

L'empresa va dur a terme un estudi previ per determinar els elements que havien d'integrar-se i usar a dins del sistema. Després de considerar totes les alternatives, s'ha escollit usar els elements/componentes següents:

1. Hardware: processador ATOM que compleix les següents característiques:
  - (a) Disponibilitat de diversos ports USB per connectar els dispositius bluetooth
  - (b) Targeta Ethernet
  - (c) *Embedded*
  - (d) Dimensions reduïdes
  - (e) Baix consum (tot lo possible)
2. Base de dades MySQL: l'empresa usa aquest motor de manera que es continuarà amb el seu ús.
3. Tecnologia Bluetooth per reunir les següents característiques:
  - (a) Baix cost
  - (b) Baix consum
  - (c) Estàndard

4. Servidor web amb suport PHP: continuant amb els llenguatges usats a l'empresa per les seves aplicacions, s'haurà de fer servir PHP.
5. Sistema operatiu Ubuntu.
6. Llenguatge C i C-Shell pel desenvolupament del sistema: La implementació del sistema (excloent la interfície gràfica i alguns programes de control) ha estat desenvolupat en codi C per raons històriques (llenguatge natiu del S.O.). A més a més, era la millor alternativa en el mercat, ja que Java té grans limitacions en l'ús dels perifèrics hardware. La pila Bluetooth del sistema operatiu Ubuntu ha estat implementada en C, fet que facilita la programació del nostre codi.
7. Els programes de control han estat programats usant C-Shell degut a què la família *Bash* és la més útil a l'hora de realitzar operacions rutinàries. Dins de Bash, s'ha escollit el C-Shell per la seva proximitat amb el codi C.

La utilització dels ports USB es necessària per tal de poder connectar els dispositius Bluetooth. La targeta Ethernet per donar-li accés a la xarxa de manera que el sistema pugui ser administrat remotament. Els altres requeriments venen de la necessitat de que el sistema sigui col·locat el més proper al públic possible així que com més petit sigui, millor. El baix consum no és indispensable però sempre abaratirà els costos de manteniment (tot i que podrien arribar a ser inapreciables).

La solució escollida suporta més característiques degut a que es possible que en un futur es desenvolupins noves funcions. El seu preu és més car per les noves propietats. L'aparell és un eBOX530 (veure figura 3.1).

Aquest hardware no és fix i pot ser reemplaçat per un altre que s'ajusti més a les condicions en cas de que s'arribi a fer servir exclusivament per a la publicitat.

Una de les restriccions inicials era utilitzar MySQL com a base de dades (la qual requereix d'una llicència comercial). En un futur, seria convenient estudiar alternatives ja que, per exemple, en el cas de PostgreSQL, la llicència no té cap cost.

El projecte ha estat desenvolupat en Ubuntu pel material ja existent que ha ajudat a portar a terme el projecte. Ubuntu es una distribució Linux basada en Debian



Figura 3.1: Hardware on s'instal·larà el sistema

GNU/Linux molt enfocada a la facilitat d'ús. La majoria del seu codi està basat en llicències lliures o bé de codi obert, cosa que suposarà una avantatge important per poder observar les implementacions existents que estiguin relacionades amb el projecte.



# Capítol 4

## Estat de l'art

En la primera part del projecte, s'ha de fer una cerca a l'entorn per detectar qui té un dispositiu Bluetooth dins del rang d'efecte. Seguidament, un cop localitzats els objectius, es fa un accés al dispositiu mòbil per conèixer les característiques dels protocols implementats allà. En el nostre cas, haurem de fer algunes comprovacions contrastant la informació obtinguda amb la nostre base de dades i per acabar s'ha de fer l'enviament de la publicitat en sí.

### 4.1 Entorn

#### 4.1.1 Hardware

El sistema requeria d'usar un sistema empotrat. Els sistemes empotrats estan dissenyats per realitzar una o poques funcions dedicades, habitualment en temps real. Normalment, el ús que se li dóna no té res a veure amb el més corrent. La majoria dels seus components són incluits dins de la placa base i moltes vegades no tenen la forma convencional (torre i pantalla) ja que solen ser molt més petits.

#### 4.1.2 Sistema operatiu

Existeixen varies alternatives al mercat. Principalment, les que dominen són Windows i GNU/Linux. La primera es software propietari el qual no permet accedir

al codi font i l'altre és majoritàriament *open-source*.

### 4.1.3 Base de dades

El motor de la base de dades escollit ha estat el MySQL. Existien diverses alternatives al mercat:

- SQL
- MySQL
- PostgreSQL

SQL és software propietari i també el més car. De les dues opcions restants, no hi ha gaire diferència entre una o l'altra ja que si bé històricament hi havia grans diferències (MySQL era molt més ràpid mentre que PostgreSQL implementava moltes més funcionalitats), a dia d'avui són molt similars.

## 4.2 Bluetooth

El Bluetooth serà la eina principal per a que s'aconsegueixin els objectius presentats. Es una tecnologia de comunicacions que té un abast curt amb la finalitat principal de reemplaçar cables a la vegada que manté alts nivells de seguretat. Els seus punts forts són el baix consum aconseguit així com els seus preus irrisoris. La seva especificació ha estat adaptada arreu del món i, actualment, gairebé qualsevol dispositiu bluetooth es pot connectar amb un altre.

### 4.2.1 Breu repàs a la història del Bluetooth

L'origen del seu nom ve d'un rei danès que va unificar tribus daneses, sueques i noruegues el qual es deia Harold Bluetooth. L'any 1994 la companyia Ericsson va iniciar varies investigacions amb la finalitat d'aconseguir una interfície a baix cost i consum per aparells electrònics (principalment, telèfons). L'any 1999 es va crear SIG, que consistia en la unió de diverses empreses (Intel, Nokia, Toshiba, IBM, Microsoft...) on el projecte va ser acabat.

### 4.2.2 Detalls tècnics

Les comunicacions són realitzades per radiofreqüència aconseguint que els dispositius no hagin d'estar alineats per a ser comunicats. Així, poden existir medis físics entre dos aparells i aquests seran capaços de comunicar-se depenent de la potència de transmissió que tinguin. Existeixen 3 classes estàndards diferents a dia d'avui (amb la possibilitat de modificacions futures ja que el pròxim estàndard Bluetooth està a punt de ser establert). Això no vol dir que les diferents classes siguin excloses entre elles ja que és possible la comunicació heterogènia.

Podem observar a la taula 4.2.2 les característiques pròpies de cada classe.

Classe	Potència màxima permesa(mW)	Rang aproximat (m)
1	100	100
2	2,5	25
3	1	1

Taula 4.1: Diferència entre classes

Cal mencionar que aquest és el model teòric. A la pràctica, l'abast dels dispositius dependrà en gran part dels elements físics existents entre els dos punts a comunicar que poden menguar la capacitat comunicativa, així com de la classe més alta existent en el procés. Per exemple, dos dispositius de classe 2 poden estar massa separats per a intercanviar dades però si tenim un de classe 1 amb un de classe 2, aquest últim podrà funcionar gràcies a que el primer dispositiu arribarà a ell i que quan li toqui enviar dades, si ve la classe 2 no arriba fins tan lluny, el de classe 1 serà capaç d'interpretar la informació ja que el seu abast és més elevat.

També existeix una classificació paral·lela segons l'ampla de banda disponible. Podem apreciar com s'implementaran grans diferències en un futur pròxim observant la taula 4.2.2.

### 4.2.3 Més dades tècniques

El rang de tràfic en el que treballa es troba entre 2,4 i 2,48GHz amb salts de freqüència per evitar interferències i d'esvaïment, així com la possibilitat de trans-

Versió	Ampla de banda (MBit/s)
1.2	1
2	3
UWB BT (proposat)	de 53 a 480

Taula 4.2: Diferència entre versions

metre en Full Duplex fins a un màxim de 1600 salts/s. (79 freq en intervals de 1Mhz). Està format per dos parts:

- El dispositiu de radio encarregat de transmetre i modular la senyal i,
- Una unitat de control de l'enllaç, per la gestió d'aquest i les funcions del software

El seu tamany es de 9x9mm.

Quan es fa referència a bluetooth, es sol parlar de piconets. Aquests són els dispositius connectats a través d'aquesta tecnologia variant el número de piconets possibles de 2 fins a 8. Cal destacar que les comunicacions poden ser punt a punt o bé punt a multipunt.

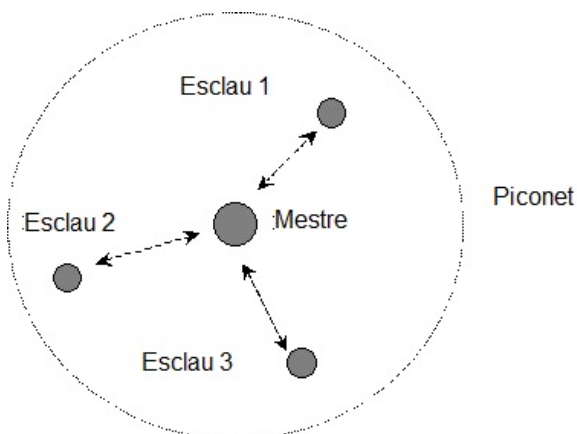


Figura 4.1: Xarxa - piconets

En el cas que es vulgui realitzar una comunicació, un dispositiu haurà de suportar certs perfils, corresponents als serveis a usar. Aquests són descripcions de



comportament general que els dispositius poden utilitzar per comunicar-se, formalitzats per afavorir un ús unificat. La forma en la que funciona per la tecnologia es basa en els perfils que suporta cada dispositiu. A continuació, veiem una llista de perfils implementats per Bluetooth:

- Advanced Audio Distribution Profile (A2DP)
- Audio/Video Remote Control Profile (AVRCP)
- Basic Imaging Profile (BIP)
- Basic Printing Profile (BPP)
- Common ISDN Access Profile (CIP)
- Cordless Telephony Profile (CTP)
- Device ID Profile (DID)
- Fax Profile (FAX)
- Dial-up Networking Profile (DUN)
- File Transfer Profile (FTP)
- General Audio/Video Distribution Profile (GAVDP)
- Generic Access Profile (GAP)
- Generic Object Exchange Profile (GOEP)
- Hard Copy Cable Replacement Profile (HCRP)
- Hands-Free Profile (HFP)
- Human Interface Device Profile (HID)
- Headset Profile (HSP)
- Intercom Profile (ICP)

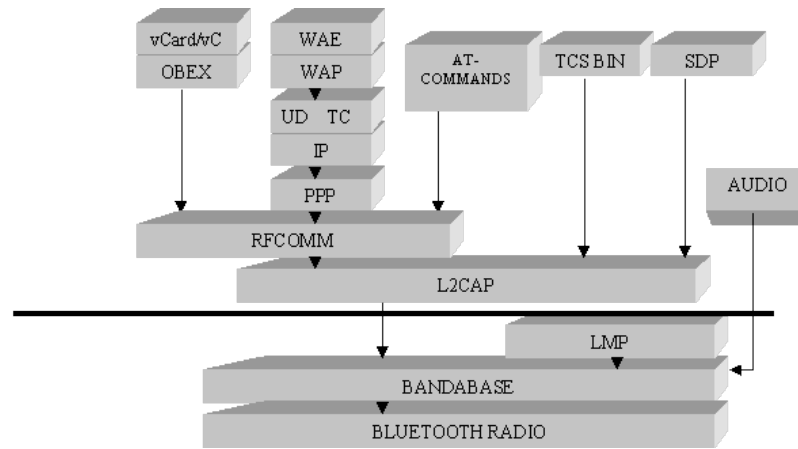


Figura 4.2: Protocols Bluetooth

- Object Push Profile (OPP)
- Personal Area Networking Profile (PAN)
- Phone Book Access Profile (PBAP)
- Serial Port Profile (SPP)
- Service Discovery Profile (SDAP)
- SIM Access Profile (SAP, SIM)
- Synchronisation Profile (SYNCH)
- Video Distribution Profile (VDP)
- Wireless Application Protocol Bearer (WAPB)

De tota aquesta llista, ens seran útils el OPP, SDAP i HID, els quals explicarem més tard.

Veiem la figura 4.2 per observar la interacció entre protocols:

Podem separar aquesta munt de protocols en 4 capes diferents:

- Protocols Bluetooth centrals (Bluetooth Core Protocols: BaseBand, LMP, L2CAP, SDP)

- Protocols reemplaçament de cable (Cable Replacement Protocols: RFCOMM)
- Protocols de control de telefonia (Telephony Control Protocols: TCS Binary, AT-Commands)
- Protocols adaptats (Adapted Protocols: PPP, UDP/TCP/IP, OBEX, WAP, vCard, vCal, IrMC, WAE)

#### 4.2.4 Bluez

Per poder gestionar les comunicacions necessàries es farà servir la pila del protocol Bluetooth oficial de Linux. És un projecte obert distribuït amb llicència GPL el qual ha sigut dissenyat modularment. Serà la capa que establirà contacte entre el sistema operatiu i la implementació del sistema a alt nivell. Actualment, és la única pila de Bluetooth que s'usa. Primerament va aparèixer la openBT (desenvolupat per Axis Communications) però va ser discontinuat al aparèixer aquesta nova pila com a millor alternativa.

Dins de Bluez podem trobar varies utilitats que ens serviran d'ajuda. Algunes han sigut implementades de nou en el projecte afegint optimitzacions.



# Capítol 5

## Disseny i implementació

### 5.1 Diagrama del sistema

El sistema ha sigut distribuït en 3 fases operatives actives i una quarta de manera passiva. Ha sigut pensat per a ser totalment autònom i no necessitar d'un manteniment continu. L'administrador hauria d'accedir el mínim de cops possible ja que no es necessari que tingui coneixements tècnics per a ser consumidor del sistema. Podem veure el diagrama de flux que té l'algorisme implementat a la figura 5.1.

Com podem observar, aquest no és complex. El sistema comprova en un inici si ha d'estar encès o parat. En cas d'estar encès i que es vulgui tenir parat, s'enviarà una senyal als processos que s'estan executant per tal de parar-los. En cas de què estigui parat i el vulguem encendre, el sistema llançarà els processos necessaris per a la gestió del sistema. Podem observar que hi ha 3 fases diferents (les quals detallarem més endavant en aquest capítol). Totes elles s'executaran i reconeixeran si els hi arriba un senyal d'aturada. En cas que si, s'atura i eliminar l'execució d'aquests processos i tornem a inicialitzar al sistema, comprovant si volem que es torni encendre o no.

Es poden tenir diversos *dongles* connectats al sistema per tal d'ampliar la seva capacitat productiva. Com més n'hi hagin, més capacitat d'acció es tindrà en entorns molt variants ja que es podran fer més comunicacions en paral·lel.

A continuació, es detallaran les diferents fases en més profunditat.

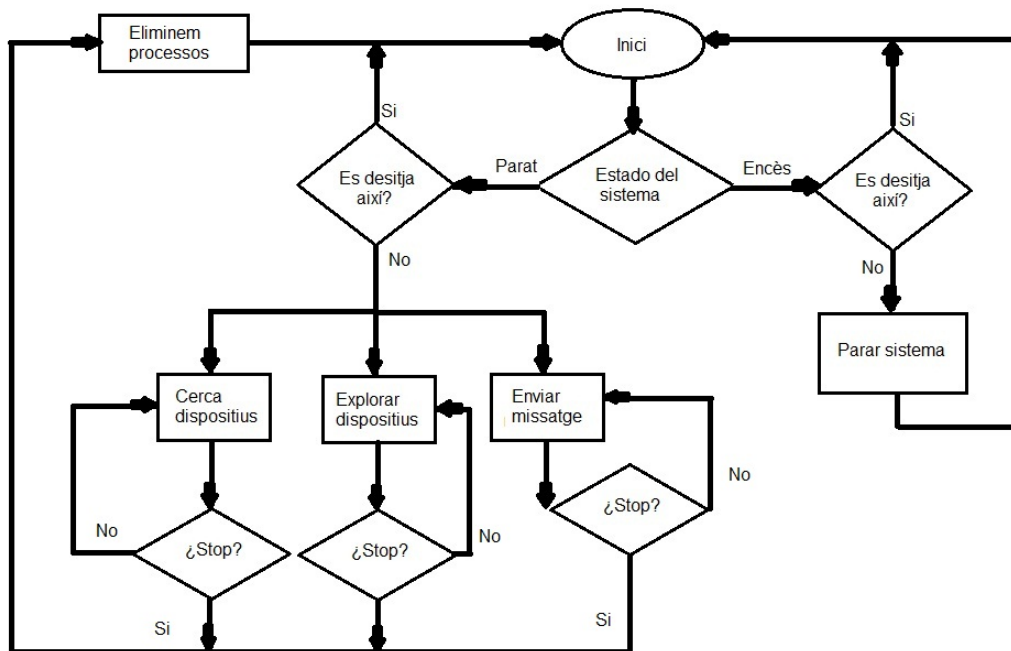


Figura 5.1: Diagrama de flux del sistema

## 5.2 Llibreries necessàries

Per tal de dur a terme el desenvolupament de projecte i assegurar una integritat total a l'hora de posar-lo en funcionament, és necessari instal·lar les següents llibreries:

Es poden instal·lar usant la comanda habitual d'Ubuntu:

```
sudo apt-get install <PAQUET_LLIBRERIA>
```

- C-Shell (csh): necessari per a l'execució d'aquest tipus d'scripts.
- Php5-gd: necessari per a la creació de gràfiques.
- Libbluetooth3-dev: permet desenvolupar aplicacions per interactuar amb el Bluetooth
- Libmysqld-dev: API MySQL per C. Gràcies a aquesta es poden fer funcions en C amb destinació al motor de bases de dades.

- Libmysqlclient16: similar a libmysqld-dev.
- Mysql-server: servidor MySQL.
- Mysql-client: client MySQL.
- Apache: servidor PHP.
- Bluez: pila Bluetooth per al sistema operatiu Ubuntu.

## 5.3 Base de Dades

Per tal de poder treballar i gestionar la informació necessària, ha sigut necessari la utilització d'una base de dades que ens permeti guardar i actualitzar les dades que usem.

### 5.3.1 Requeriments de dades

A l'hora de crear una base de dades, el primer pas és decidir quina informació ens farà falta.

Primer de tot, és necessari guardar la direcció MAC dels dispositius locals per tal de poder guardar la seva direcció física i utilitzar-los. A més a més, serà necessari guardar el número assignat pel processador a aquests ja que segons el programa, treballarà amb la MAC o bé amb la seva numeració. Per altre banda, tenim un administrador que s'haurà de identificar per accedir al sistema. No és necessari guardar molta informació al respecte ja que no es pretén tenir informació sobre l'administrador. En saber el seu nom (per tenir una salutació quan s'identifiqui), el DNI (que serà el seu nom d'usuari) i la paraula de pas, serà suficient. També guardarem el rang que aquest pot tenir pensant en una millora futura (veure conclusions) però per ara no és essencial.

I les dades més importants: la dels dispositius cercats, els canals dels seus protocols... La informació bàsica són les MACs remotes degut a què sinó no es podria establir cap tipus de comunicació. A més a més, per a cada fase, s'ha de guardar el temps en el què s'ha realitzat la última acció ja que això ens permetrà

tenir un control de prioritats més tard. Hem de saber quin és el canal que el SDPTOOL ens retornarà i el número d'intents que s'han fet (de nou, per establir prioritats). El mateix passa per al procés d'enviar la publicitat: hem de saber si ja s'ha realitzat o no. També s'hauria de guardar el nom de la última imatge enviada ja que en un canvi de publicitat, s'ha de tornar a contactar amb el telèfon en cas de ser possible.

Encara són necessàries més dades: la llista negra. Com explicarem en l'apartat 3.7.5, és necessari guardar una llista de MACs amb les quals el sistema no pugui interactuar.

Per últim, per les rutines de control dels scripts ens farà falta saber els processos que s'estan executant, la seva identificació en el sistema (PID) i a quina fase correspon aquell procés. També serà necessari saber quan va iniciar-se ja que algunes rutines forçaran un time out.

### 5.3.2 Disseny lògic

En la figura 5.2 podem veure el disseny de la base de dades.

#### Llistat de les taules

- T. *Dongles*: en aquesta taula es guardarà tota la informació referent als processos que s'estan executant i serà de suport per les rutines de control.
  - MAC [Caràcters](17): guardarà la direcció física dels dispositius. Serà la clau primària del sistema perquè no tindrem mai dos dispositius amb la mateixa MAC.
  - Tipus [Numèric]: tindrà el tipus d'acció a realitzar (a quina fase del procés serà usat).
  - HCI [Numèric]: número del dispositiu assignat pel processador.
- T. *Users*: informació relativa a l'administrador.
  - Nom [Caràcters](17): guardarà el nom de l'administrador.



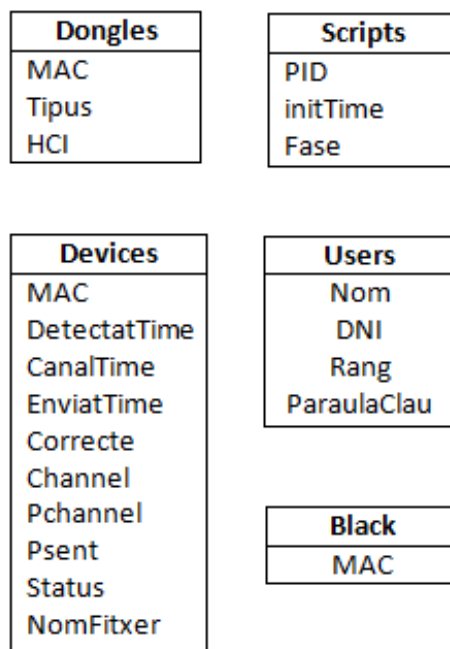


Figura 5.2: Disseny lògic de la base de dades

- DNI [Numèric]: identificació nacional de l'administrador. Serà el seu nom d'usuari per accedir al sistema ja que es únic (i per tant, clau primària).
- ParaulaClau [Caràcters] (20): contrasenya per a accedir al sistema.
- Rang [Numèric]: quin serà el rang de domini d'aquest administrador.
- T. Black: aquí tindrem la informació necessària per negligir l'enviament de publicitat a mòbils coneguts.
  - MAC [Caràcters](50): guardarà les direccions físiques bloquejades. Com només hi pot haver una instanciació per telèfon, serà clau primària.
- T. Devices: aquí es guardarà tota la informació relativa a les diferents fases del sistema. La informació aquí guardada s'actualitzarà constantment.
  - MAC [Caràcters](17): guardarà la direcció física dels dispositius.

- DetectatTime [Data]: registre del moment en el què s’ha detectat un nou dispositiu.
  - CanalTime [Data]: registre del temps en el què se li ha buscat el canal a aquest nou dispositiu
  - EnviatTime [Data](17): registre de l’hora en la que s’ha enviat un missatge.
  - Correcte [Numèric]: ens servirà de variable de control per a la fase 3.
  - Channel [Numèric]: canal del perfil OOP que el dispositiu remot té assignat.
  - Pchannel [Numèric]: variable de control per a la fase 2 per saber si ja tenim el canal o està pendent de ser explorat.
  - Psent [Numèric]: variable de control per a la fase 3 que ens indica si ja s’ha enviat a la MAC corresponen o no.
  - Status [Numèric]: variable especial per funcions extres.
  - NomFitxer [Caràcters] (60): non de l’últim fitxer que s’ha enviat a l’objectiu.
- T. Scripts: les dades necessàries per al control de rutines estaran definides aquí.
    - PID [Numèric](17): número d’identificació de procés atorgat pel processador a algun dels programes en funcionament.
    - initTime [Data]: moment en el què el PID ha sigut iniciat.
    - Fase [Numèric]: a quina de les fases correspon el PID.

## 5.4 Fase 1 - Cerca

En el capítol de l’estat de l’art hem parlat dels diferents perfils que implementa el Bluetooth per tal de comunicar-se. En aquest cas, hem de fer servir el *HCITool* que serveix per configurar les connexions bluetooth així com enviar comandes

definides als dispositius que estan dins del radi d'acció. El seu funcionament és senzill: obre una connexió per socket amb l'especificació *AF\_BLUETOOTH* i *BTPROTO\_HCI* per tal de poder establir la comunicació entre els dispositius i es fa una crida *IOCTL*<sup>1</sup> indicant la comanda que volem executar; en aquest cas, un *HCINQUIRY*. Ens servirà per fer un reconeixement de l'entorn i capturar les direccions físiques dels telèfons objectius.

El procés de *inquiry* no sempre funciona en tots els mòbils: han d'estar en mode visible. En cas que ho estiguin, en el moment que el nostre dispositiu envia la comanda, un packet que s'anomena *inquiry state*, per ràdio (similar al funcionament d'un *broadcast*), tots aquells que la rebin i estiguin activats en el *inquiry scan state* (ja que sinó, no reaccionaràn) canviaran el seu estat a *inquiry response state* i enviaran un paquet de resposta al emissor.

Existeix una comanda en Bluez que implementa *HCITool* però com veurem ara, no és completa i ha hagut de ser millorada.

La sintaxis és la següent:

```
hcitool [-i <hciX>] [command [command parameters]]
```

On *hciX* correspon al dongle que tinguem connectat i *command* es quina de les operacions disponibles en el *HCITool* durem a terme. En el nostre cas, la crida es la següent:

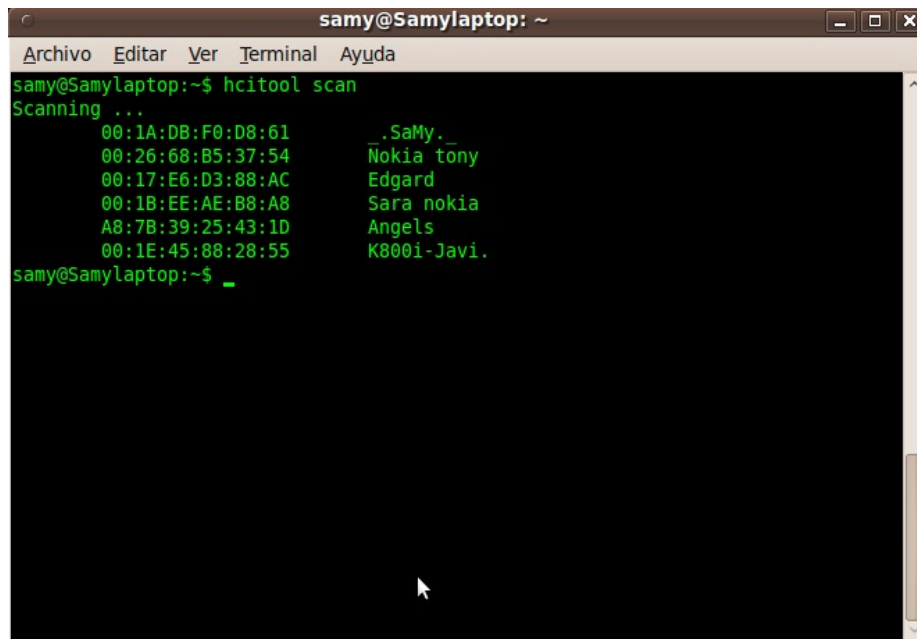
```
\$hcitool -i <MAC_DONGLE> scan
```

Quan executem la comanda superior ens es retornat una llista dels dispositius Bluetooth (mòbils presumiblement) amb la seva identificació MAC i també, el nom que cada usuari ha pogut personalitzar. Podem veure una captura en la figura 5.3:

En principi, tota la informació que volíem està aquí: tenim una llista dels dispositius trobats amb les seves direccions per a poder interactuar amb ells. Tot

---

<sup>1</sup>*IOCTL* es una crida al sistema que permet a una aplicació controlar o comunicar-se amb un driver (en aquest cas, del bluetooth, prèviament cridat pel socket). Cada driver ja tindrà definit la seva actuació segona la crida *IOCTL* establerta.



```
samy@Samylaptop: ~  
Archivo  Editar  Ver  Terminal  Ayuda  
samy@Samylaptop:~$ hcitool scan  
Scanning ...  
    00:1A:DB:F0:D8:61      _SaMy._  
    00:26:68:B5:37:54      Nokia tony  
    00:17:E6:D3:88:AC      Edgard  
    00:1B:EE:AE:B8:A8      Sara nokia  
    A8:7B:39:25:43:1D      Angels  
    00:1E:45:88:28:55      K800i-Javi.  
samy@Samylaptop:~$ _
```

Figura 5.3: Cerca de dispositius usant mètode per defecte

i així, aquesta instrucció presenta certs inconvenients que van impossibilitar que es pogués utilitzar exclusivament aquesta opció. El primer és el fet de no poder executar aquest procés en dos *dongles* locals a la vegada ja que els recursos del processador es troben ocupats i per tant, no es podia aplicar concurrència. A més, per defecte (i sense que es pugui canviar), aquesta comanda no netejaria un *flag* del nostre dispositiu anomenat *IREQ\_CACHE\_FLUSH* que s'encarrega de netejar la memòria cau. *¿Quines conseqüències té això?* Quan tornem a llançar el programa, degut a què el *flag* està activat a 0, els resultats de cerques anteriors són retornats malgrat la comunicació ja no sigui possible perquè han deixat d'estar dins del rang de cobertura. Això provoca que els nostres dispositius dedicats a la cerca no siguin útils durant un període massa gran de temps i faci molt ineficaç el sistema.

### Solució implementada

En lloc d'utilitzar aquesta instrucció i gràcies a què el codi en C del *HCITool* està disponible, s'ha implementat una versió millorada la qual neteja el *flag* de la memòria cau cada vegada que fa una cerca. D'aquesta manera, cada cop que s'executa el programa no parteix d'una base d'informació existent sinó que conté una llista buida d'elements i per tant, tots els que retorna estan dins del rang de cobertura sempre. Per acabar-ho d'optimitzar, per cada dispositiu local es crea un procés diferent fent possible la concurrència.

Cal destacar alguns detalls tècnics sobre aquest procés per a la seva configuració. Primerament, s'ha definit que retorni un màxim de 255 dispositius (és a dir, que si hi haguessin 300 disponibles, 50 serien omesos) degut al fet que està molt recomanat de manera estàndard per a no sobresaturar el servidor o bé que hi hagi un overflow per no poder administrar tantes dades. També cal considerar el fet del temps necessari per a dur a terme 1 cerca: Per cada byte rebut, hi ha un retard de 1,28s. En el nostre cas hem de rebre 8 bytes (48 bits que ocupa la direcció física) així que el temps per cerca es, obligatòriament i de forma mínima 10,24 segons.

Tot i així, com hem mencionat abans, la instrucció definida també retornava, a més a més de la direcció MAC, el nom *friendly* que cadascú pot personalitzar en el seu mòbil per a les connexions Bluetooth. Si una cosa del sistema es crítica, aquesta és el temps d'acció per a enviar publicitat degut a què s'actuarà en un entorn molt variant i s'ha d'actuar el més ràpid possible. Guardar el nom *friendly* podria ser útil a l'hora d'integrar-lo a la interfície gràfica ja que seria més vistós pel client però, ¿realment és útil? No es informació rellevant ni es farà servir en cap altre moment del procés així que si suposava un cost (en temps) elevat, una bona optimització seria suprimir aquest segon procés per tal no perdre aquest temps en les cerques. Els resultats de les proves han sigut recollits a la taula 5.1.

Com podem veure, el fet d'afegir una funcionalitat poc útil al sistema té un cost computacional molt alt ja que el retorn de les dades es prolonga durant molt més temps: gairebé el doble que l'estipulat. També podem observar com en el procés de cerca no hi ha variacions en el temps necessari per a la comunicació i que es pràcticament el definit per estàndard. La diferència de 10,24 a 10,28 es

Amb segon procés (segons)	Únicament cerca per la MAC (segons)
15.1	10.28
25.6	10.28
17.3	10.28
21.3	10.29
14.9	10.29
18.5	10.29
Mitjana: 18,78	Mitjana: 10.28

Taula 5.1: Diferència entre implementacions

deurà a un petit overhead per a la gestió dels recursos.

Concloues aquestes proves, es va decidir eliminar totalment el segon procés per a fer un sistema que només busqués MACs.

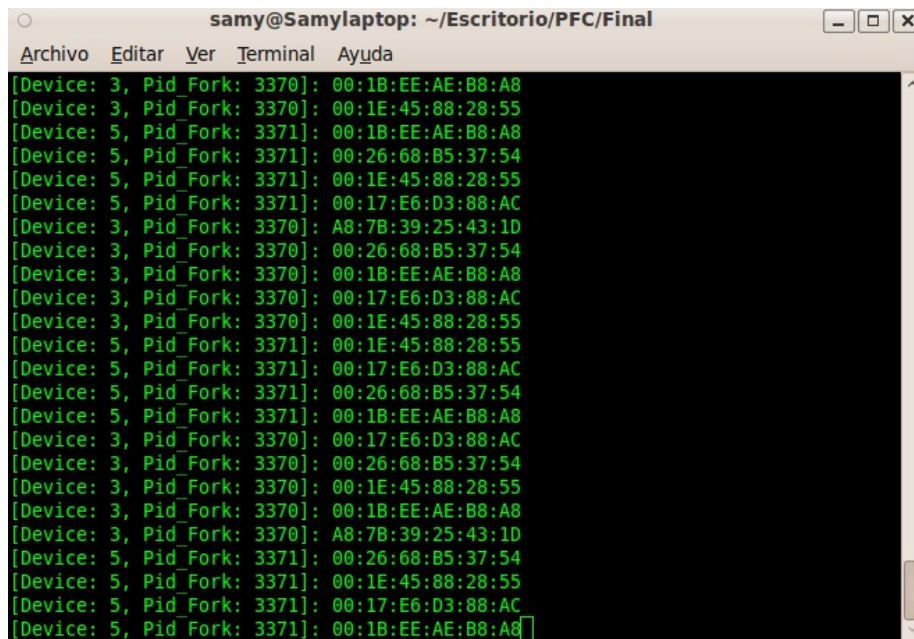
### Algorisme

L'algorisme es trivial: selecciona els dispositius destinats a la cerca segons la nostre base de dades i per a cada un es crea un procés independent (ja que sinó es bloquegen els recursos destinats a la comunicació Bluetooth i no podem fer cerques paral·leles) que iniciarà la seva activitat. Un cop els resultats són retornats, s'insereixen a la base de dades corresponent. Podem eure la sortida per pantalla d'aquest codi a la figura 5.4.

Per últim, cal destacar la freqüència amb la que es fan cerques: Sabem que cada cerca dura 10 segons i aquest temps es fix. Per tant, hem de buscar un número òptim de dispositius dedicats a aquesta fase. Lo ideal es una nova cerca cada 3 ó 5 segons degut al fet que si bé podem posar que tinguem una de nova cada segon, aquesta sortirà moltes vegades repetides i estarà utilitzant *dongles* inútilment. En canvi, com més en dediquem a l'enviament dels fitxers, més públic a la vegada rebrà un missatge i per tant, hi haurà més possibilitats d'èxit.

L'equació era molt simple:

$$10 = [3 - 5] * NumDongles \quad (5.1)$$



```

samy@Samylaptop: ~/Escritorio/PFC/Final
Archivo  Editar  Ver  Terminal  Ayuda
[Device: 3, Pid_Fork: 3370]: 00:1B:EE:AE:B8:A8
[Device: 3, Pid_Fork: 3370]: 00:1E:45:88:28:55
[Device: 5, Pid_Fork: 3371]: 00:1B:EE:AE:B8:A8
[Device: 5, Pid_Fork: 3371]: 00:26:68:B5:37:54
[Device: 5, Pid_Fork: 3371]: 00:1E:45:88:28:55
[Device: 5, Pid_Fork: 3371]: 00:17:E6:D3:88:AC
[Device: 3, Pid_Fork: 3370]: A8:7B:39:25:43:1D
[Device: 3, Pid_Fork: 3370]: 00:26:68:B5:37:54
[Device: 3, Pid_Fork: 3370]: 00:1B:EE:AE:B8:A8
[Device: 3, Pid_Fork: 3370]: 00:17:E6:D3:88:AC
[Device: 3, Pid_Fork: 3370]: 00:1E:45:88:28:55
[Device: 5, Pid_Fork: 3371]: 00:1E:45:88:28:55
[Device: 5, Pid_Fork: 3371]: 00:17:E6:D3:88:AC
[Device: 5, Pid_Fork: 3371]: 00:26:68:B5:37:54
[Device: 5, Pid_Fork: 3371]: 00:1B:EE:AE:B8:A8
[Device: 3, Pid_Fork: 3370]: 00:17:E6:D3:88:AC
[Device: 3, Pid_Fork: 3370]: 00:26:68:B5:37:54
[Device: 3, Pid_Fork: 3370]: 00:1E:45:88:28:55
[Device: 3, Pid_Fork: 3370]: 00:1B:EE:AE:B8:A8
[Device: 3, Pid_Fork: 3370]: A8:7B:39:25:43:1D
[Device: 5, Pid_Fork: 3371]: 00:26:68:B5:37:54
[Device: 5, Pid_Fork: 3371]: 00:1E:45:88:28:55
[Device: 5, Pid_Fork: 3371]: 00:17:E6:D3:88:AC
[Device: 5, Pid_Fork: 3371]: 00:1B:EE:AE:B8:A8

```

Figura 5.4: Cerca de dispositius versió millorada

Si disposem de pocs *dongles*, en destinarem 2 a la cerca. En cas de tenir una capacitat més alta, 3 serà el número destinat a aquesta fase.

Aquest procés és totalment transparent ja que l'usuari no té la capacitat per percebre que el seu telèfon ha sigut cercat degut a què les respostes als paquets enviats son automàtiques. A continuació, veiem l'estat de la base de dades a la figura 5.5 un cop acabada aquesta fase:

## 5.5 Fase 2 - Exploració

En aquesta fase farem servir un altre dels perfils disponibles: el *SDPTOOL*.

*SDPTOOL* és una utilitat que ens permet saber quins serveis estan disponibles en un dispositiu Bluetooth i quines característiques té. Així, ens pot interessar saber per quin canal funciona un mans lliures o bé, per on es comunicarà a l'hora de fer una transferència de fitxers.

De nou, Bluez porta incorporat les eines per a fer servir *SDPTOOL*. En aque-

```
mysql> select * from dev;
```

mac	detected	canaltime	enviado	correcto	channel	pchannel	psent	status	fitxer
A8:7B:39:25:43:1D	2010-06-21 03:03:24	0000-00-00 00:00:00	0000-00-00 00:00:00	0	-1	0	0	0	NULL
00:17:E6:03:88:AC	2010-06-21 03:03:24	0000-00-00 00:00:00	0000-00-00 00:00:00	0	-1	0	0	0	NULL
00:1B:EE:AE:B8:A8	2010-06-21 03:03:24	0000-00-00 00:00:00	0000-00-00 00:00:00	0	-1	0	0	0	NULL
00:26:68:85:37:54	2010-06-21 03:03:24	0000-00-00 00:00:00	0000-00-00 00:00:00	0	-1	0	0	0	NULL
00:1E:45:88:28:55	2010-06-21 03:03:24	0000-00-00 00:00:00	0000-00-00 00:00:00	0	-1	0	0	0	NULL

```
5 rows in set (0,00 sec)

mysql>
```

Figura 5.5: Base de dades després d'aplicar una cerca de dispositius

st cas, estan molt més desenvolupades i és possible filtrar tota la informació innecessària per obtenir només allò que necessitem de manera que el programa ja existent per a aquest perfil correctament configurat ens donarà el resultat esperat sempre.

La sintaxis d'aquesta instrucció és la següent:

```
sdptool [options] {command} [command parameters ...]
```

En el nostre cas, usarem la opció destinada a escollir quin dels *dongles* usarem (per a poder seleccionar els que haguem definit per a aquesta operació). També indicarem que volem fer una exploració en un determinat dispositiu i filtrarem per a què la seva sortida retorni, únicament, la informació del perfil què ens interessa. La comanda final tindrà el següent format:

```
sdptool $-i <MAC_DONGLE> search --bdaddr <MAC_OBJECTIU> 0x1105
```

L'últim paràmetre de tots correspon a la codificació del protocol que ens interessa explorar: el OOP. Ens donarà una llista de detalls tècnics del qual ens es necessari només el canal per el que es farà la connexió. Si en un determinat dispositiu que implementa el protocol OOP fa servir el canal 4, només hi podrem accedir, exclusivament, per aquest. Podem veure l'actuació d'aquest procés en la figura 5.6.



```

samy@Samylaptop: ~/Escritorio/PFC/Final
Archivo Editar Ver Terminal Ayuda

[DEBUG] -> Dos MACs a ser mandadas! 00:1B:EE:AE:B8:A8 y 00:1E:45:88:28:55.
[DEBUG] -> MAC 0: 00:26:68:B5:37:54. MAC 1: 00:1B:EE:AE:B8:A8. Les dos tenen un channel -1
[DEBUG] -> Dos MACs a ser mandadas! 00:26:68:B5:37:54 y 00:1B:EE:AE:B8:A8.
[DEBUG] -> Hijo 1: 4403. ./canal.csh 00:80:5A:63:00:63 00:26:68:B5:37:54. ./canal.csh 00:80:5A:63:00:46 00:1B:EE:AE:B8:A8INFO: Subscript out of range.
INFO: Subscript out of range.

[DEBUG] -> Dos MACs a ser mandadas! 00:26:68:B5:37:54 y 00:1B:EE:AE:B8:A8.
[DEBUG] -> MAC 0: 00:1E:45:88:28:55. MAC 1: 00:26:68:B5:37:54. Les dos tenen un channel -1
[DEBUG] -> Dos MACs a ser mandadas! 00:1E:45:88:28:55 y 00:26:68:B5:37:54. ./canal.csh 00:80:5A:63:00:46 00:26:68:B5:37:54
[DEBUG] -> Hijo 1: 4418. ./canal.csh 00:80:5A:63:00:63 00:1E:45:88:28:55INFO: Subscript out of range.
**** [CANAL] CORRECTO. Canal es: 0

[DEBUG] -> Dos MACs a ser mandadas! 00:1E:45:88:28:55 y 00:26:68:B5:37:54.
[DEBUG] -> MAC 0: 00:1B:EE:AE:B8:A8. MAC 1: 00:26:68:B5:37:54. Les dos tenen un channel -1
[DEBUG] -> Dos MACs a ser mandadas! 00:1B:EE:AE:B8:A8 y 00:26:68:B5:37:54.
[DEBUG] -> Hijo 1: 4434. ./canal.csh 00:80:5A:63:00:63 00:1B:EE:AE:B8:A8. ./canal.csh 00:80:5A:63:00:46 00:26:68:B5:37:54INFO: Subscript out of range.
**** [CANAL] CORRECTO. Canal es: 5

[DEBUG] -> Dos MACs a ser mandadas! 00:1B:EE:AE:B8:A8 y 00:26:68:B5:37:54.
[DEBUG] -> MAC 0: 00:1B:EE:AE:B8:A8. MAC 1: 00:26:68:B5:37:54. Les dos tenen un channel -1
[DEBUG] -> Solo tenemos la MAC 00:1B:EE:AE:B8:A8 esperando... Let's go. ./canal.csh 00:80:5A:63:00:63 00:1B:EE:AE:B8:A8INFO: Subscript out of range.

[DEBUG] -> Solo tenemos la MAC 00:1B:EE:AE:B8:A8 esperando... Let's go

```

Figura 5.6: Resultat per pantalla d'aplicar la fase 2

## Algorisme

L'algorisme a seguir és trivial: es seleccionen els recursos a usar que destinarem a aquesta etapa i fem consultes a la base de dades de manera contínua. Quan apareix una MAC objectiu es llança un procés en busca dels seus protocols. El cos principal està fet en C i degut a què podem fer servir una instrucció del sistema per a dur a terme l'exploració, s'ha desenvolupat aquest part en C-Shell. Així, el codi C pot arribar a llançar tants processos com dispositius associats tingui a la tasca i aquests llançaran un script. Aquest s'encarregarà de realitzar la exploració i actualitzar la base de dades amb les noves dades. Finalment, s'acaba el procés i es torna a iniciar des del inici.

En les proves realitzades, s'ha pogut observar com molts telèfons tenen problemes per respondre els paquets de petició correctament. Probablement, és el coll d'ampolla més gran ja que un cop reunides totes les dades i com explicarem al següent capítol, l'enviament sol ser eficaç (sempre que el telèfon tingui els perfils disponibles).

## Optimització de la implementació

Es va implementar una millora substancial en aquesta fase: després de varies proves, va quedar palès que si un telèfon no responia en un interval màxim de 3 segons, era altament probable que aquest ja no estigués dins del radi d'acció, entre d'altres, com parlar el bluetooth just en aquell moment o bé que el telèfon

tingués els recursos ocupats i no fos capaç d'enviar una resposta. El problema venia que el programa quedava a la espera d'una resposta en un temps de gairebé 20 segons i això suposava un coll d'ampolla massa gran per al sistema ja que tot el tràfic i operacions queda bloquejat fins que no es salta al següent objectiu (no es pot fer cap enviament si no tenim el resultat d'aquesta exploració). Per tant, paral·lelament a aquesta fase es llança un script (que explicarem més endavant) que s'encarrega de descongestionar aquest flux del sistema fent possible saltar a altres telèfons on la resposta és més probable.

L'elecció del següent objectiu es basa en una llei de probabilitat. Un exemple ho deixarà més clar:

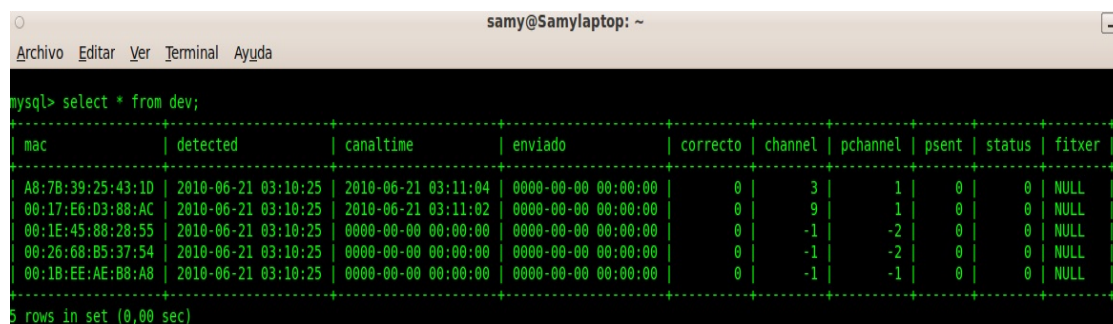
*¿Què és més probable, que dins del radi d'acció estigui un telèfon que acabem de detectar, o bé un que fa 5 minuts que hem detectat però encara no hi ha hagut temps d'explorar?*

Sembla una resposta bastant clara. Òbviament, el que acabem de detectar. Per tant, a l'hora de seleccionar un objectiu s'escollirà el més recent així com el que haguem fet menys intents d'exploració infructuoses prèviament. Cada cop que intentem obtenir resposta d'un dispositiu, en cas de rebre una resposta incorrecta, es torna a deixar en estat pendent i s'actualitza un comptador associat. Així, a més a més del temps d'antiguitat (factor principal), en cas d'empat es seleccionarà aquell amb menys intents previs ja que les opcions de que el que ha contestat més vegades malament ho torni a fer, són majors.

L'assignació de dispositius locals a aquesta fase és bastant crítica ja que com diem, es on es forma el coll d'ampolla. Conseqüentment, ha de tenir el màxim número de dispositius assignats (a la inversa del que passava en la primera fase) i està compensat amb l'enviament de missatges.

## 5.6 Fase 3 - Enviament

La última utilitat de la que farem ús és del perfil de càrrega d'objectes (Object Push Profile originalment). Aquest implementa l'ús de l'enviament de fitxers a través del protocol *OBEX* el qual permet realitzar les operacions esmentades d'un



```
mysql> select * from dev;
```

mac	detected	canaltime	enviado	correcto	channel	pchannel	psent	status	fitxer
A8:7B:39:25:43:1D	2010-06-21 03:10:25	2010-06-21 03:11:04	0000-00-00 00:00:00	0	3	1	0	0	NULL
00:17:E6:D3:88:AC	2010-06-21 03:10:25	2010-06-21 03:11:02	0000-00-00 00:00:00	0	9	1	0	0	NULL
00:1E:45:88:28:55	2010-06-21 03:10:25	0000-00-00 00:00:00	0000-00-00 00:00:00	0	-1	-2	0	0	NULL
00:26:68:B5:37:54	2010-06-21 03:10:25	0000-00-00 00:00:00	0000-00-00 00:00:00	0	-1	-2	0	0	NULL
00:1B:EE:AE:B8:A8	2010-06-21 03:10:25	0000-00-00 00:00:00	0000-00-00 00:00:00	0	-1	-1	0	0	NULL

5 rows in set (0.00 sec)

Figura 5.7: Estat de la Base de dades després d'aplicar una exploració als dispositius localitzats

dispositiu a un altre. Inicialment, era necessari associar un canal *RFCOMM* amb el telèfon abans de poder-se fer servir però ha sigut optimitzat al punt que l'usuari només ha de confirmar la recepció de les dades estalviant així passos intermedis com el PIN.

El protocol *OBEX* és de comunicacions i facilita l'intercanvi d'objectes binaris entre dispositius. És mantingut per la IrDA i per el *Bluetooth Special Interest Group*. Va ser dissenyat per a intercanviar targetes de negocis, dades e inclús aplicacions en una Palm, per arribar a expandir-se a la majoria de telèfons. El que inicia la comunicació sempre és l'emissor i fa servir una API per les operacions de connexió i desconnexió, enviament, recepció i cancel·lació. Per sobre de *OBEX*, ja trobem indirectament, l'especificació de la pila bluetooth (*Bluez*).

El seu funcionament és similar al *HTTP* però amb lleugeres diferències. El seu objectiu és oferir un transport fiable per intercanviar dades.

- La capa de transport: *OBEX* no s'implementa sobre un port TCP/IP sinó sobre una pila Banda Base/Link Manager/L2CAP/RFCOMM quan està funcionant amb bluetooth.
- Transmissió de dades: en lloc d'utilitzar text comprensible per a les persones, *OBEX* fa servir tripletes binàries anomenades headers per intercanviar informació sobre una petició u objecte. Això és degut a la falta de recursos de processament de molts dispositius amb característiques limitades.

- Sessions: una operació HTTP no té estat pròpiament. El funcionament consisteix en obrir una connexió, efectua la petició oportuna i tancar la connexió. En canvi, en *OBEX*, una sola connexió pot realitzar diverses operacions relacionades entre si. Actualment, s'està desenvolupant aquest apartat amb més profunditat afegint noves funcionalitats com l'emmagatzematge d'informació de les connexions.

En aquest cas, s'ha usat un programa ja existent anomenat *ussp-push*. La seva sintaxis es la següent:

```
ussp-push [--dev <DEVID>] {MAC_OBJECTIU@[CANAL_DESCOBERT_OOP]}
FITXER_LOCAL NOM_FITXER_REMOT
```

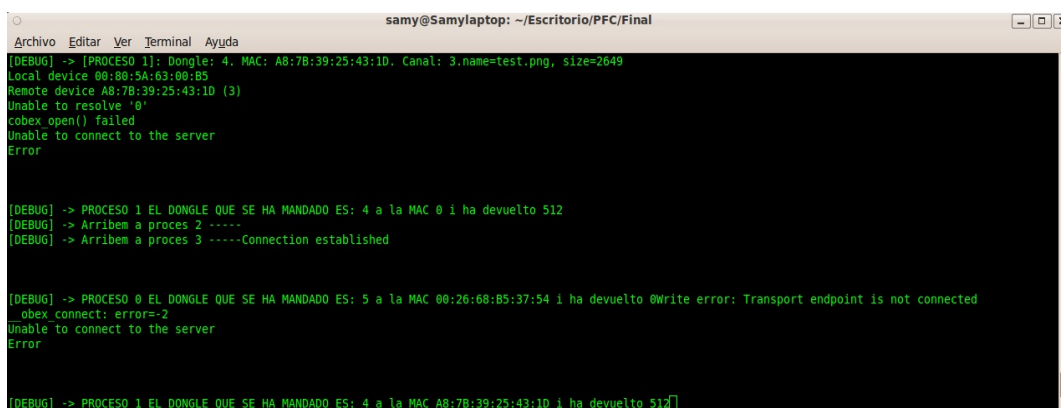
El *DEVID* correspon al número assignat a un dels nostres dispositius, la *MAC* objectiu al telèfon que volem intervenir, el canal descobert és el resultat de l'exploració i per últim, s'han d'indicar el fitxer origen així com el nom del fitxer destí.

### Algorisme

L'algorisme és el següent: Es busquen els dispositius que es dedicaran a aquesta fase (amb una breu consulta a la base de dades) i es llança un procés diferent per a cada un. Això és necessari ja que sinó hi ha conflictes de recursos i per tant, no ens permetria fer enviaments paral·lels. Gràcies a una programació concurrent s'ha pogut solucionar aquest inconvenient podent enviar a tants objectius com bluetooth locals connectats tinguem. Cada procés d'aquesta fase va comprovant les dades de la base de dades fins que troba una entrada amb l'estat d'espera per a l'enviament. Modifica el seu estat (per a evitar dos connexions amb el mateix telèfon) i prossegueix a intentar un enviament. L'usuari rebrà una petició de confirmació per rebre un missatge i si confirma, li serà enviat. En cas contrari, serà denegat i es retornarà error.

Cal destacar que en aquesta última fase, no existeix un control exhaustiu dels codis d'error retornats. En cas de time out, cancel·lació o impossibilitat de connexió, simplement es captura com a intent fallit. Una millora consistiria en fer el

codi personalitzat (en lloc d'usar aquest programa, crear-ne un que actués directament sobre la pila Blue; la idea ha sigut comentada més extensament a l'apartat de millores) i així es podrien controlar les excepcions al implementar el protocol per un mateix. Conseqüentment, depenen del problema ocasionat en l'enviament es podria pensar en fer una tanda de reintents (pels casos que l'usuari no ha vist el mòbil a la primera i a temps). La sortida per pantalla la podem veure a la figura 5.8.



```
samy@Samylaptop: ~/Escritorio/PFC/Final
[DEBUG] -> [PROCESO 1]: Dongle: 4. MAC: A8:7B:39:25:43:1D. Canal: 3.name=test.png, size=2649
Local device 00:80:5A:63:00:85
Remote device A8:7B:39:25:43:1D (3)
Unable to resolve '0'
cobex open() failed
Unable to connect to the server
Error

[DEBUG] -> PROCESO 1 EL DONGLE QUE SE HA MANDADO ES: 4 a la MAC 0 i ha devuelto 512
[DEBUG] -> Arribem a proces 2 -----
[DEBUG] -> Arribem a proces 3 -----Connection established

[DEBUG] -> PROCESO 0 EL DONGLE QUE SE HA MANDADO ES: 5 a la MAC 00:26:68:B5:37:54 i ha devuelto 0Write error: Transport endpoint is not connected
cobex_connect: error=2
Unable to connect to the server
Error

[DEBUG] -> PROCESO 1 EL DONGLE QUE SE HA MANDADO ES: 4 a la MAC A8:7B:39:25:43:1D i ha devuelto 512
```

Figura 5.8: Sortida per pantalla de la fase 3

En aquest cas, no és un procés problemàtic però la capacitat d'enviament de publicitat depèn d'ell ja que és l'últim tram on s'efectua aquesta part. Per altre banda, la fase anterior necessitava bastants recursos per a poder funciona correctament (el nombre d'intents per aconseguir les dades que volem podia ser elevat). Així, per tal d'arribar en un equilibri entre l'interès funcional (màxima velocitat) i l'interès comercial (màxim enviament concurrent possible) el número de dispositius destinats a aquest procés serà el mateix que en el d'exploració.

## 5.7 Interfície

La interfície té el requisit d'evitar problemes de compatibilitat i que sigui de fàcil accés. Per això, es va optar per a una solució via web: es pot fer servir des de qualsevol navegador. És una avantatge molt important ja que permetrà ser

```
mysql> select * from dev;
```

mac	detected	canaltime	enviado	correcto	channel	pchannel	psent	status	fixer
00:17:E6:D3:88:AC	2010-06-21 03:20:57	2010-06-21 03:21:01	2010-06-21 03:21:07	1	9	1	-1	0	NULL
00:1E:45:88:28:55	2010-06-21 03:20:57	2010-06-21 03:21:01	2010-06-21 03:21:33	1	6	1	-2	0	NULL
00:1B:EE:AE:88:A8	2010-06-21 03:20:57	2010-06-21 03:21:09	2010-06-21 03:21:16	0	9	1	1	0	NULL
A8:7B:39:25:43:10	2010-06-21 03:20:57	2010-06-21 03:21:20	2010-06-21 03:22:02	1	3	1	-1	0	NULL
00:26:68:B5:37:54	2010-06-21 03:20:57	2010-06-21 03:21:09	2010-06-21 03:21:43	0	5	1	1	0	NULL

```
5 rows in set (0.00 sec)
```

Figura 5.9: Estat final després de tots els processos

independent del software i del hardware des d'on es vulguin consultar o modificar paràmetres del sistema.

Partint d'aquesta base i degut a què no cal una seguretat extrema ni es necessiten de característiques fora de lo corrent, s'ha dissenyat una interfície gràfica en PHP/HTML/CSS/JS. S'ha instal·lat un servidor web en el hardware que permetrà l'accés en LAN (en un principi) al sistema i serà configurat des d'allà. Els canvis de publicitat, (des)activació del servei i demés, seran variables controlades per la interfície.

Existeixen diferents pàgines, així que farem una breu explicació de cada una per a poder veure les seves característiques. Això sí, sempre tindrà la opció de sortir del sistema. A més, a l'entrada serà saludat amb el seu nom.

### 5.7.1 La pàgina d'identificació

Servirà per a donar accés a l'administrador del sistema un cop introdueixi nom d'usuari i contrasenya (figura 5.10). Es crea una sessió amb cookies d'un màxim de 30 minuts, temps a partir del qual s'haurà de torna a identificar per a seguir configurant el sistema.

### 5.7.2 El resum general

En aquesta part es veu una llista d'informació general per saber com està funcionant el sistema. És la primera pàgina que es veu un cop t'has identificat. Les

The image shows a login interface for a system called 'sistemas Nhashi'. At the top left is a logo with the letters 'h' and 'n' in black and red. To its right is the text 'Bienvenido a sistemas Nhashi'. Below this is a section titled 'Introduzca sus datos' enclosed in a dashed box. Inside this box are two input fields: the first is labeled 'Nombre de usuario' with a person icon and contains the text 'Su nombre'; the second is labeled 'Contraseña' with a lock icon and is empty. To the right of these fields is a button labeled 'ENTRAR'. At the bottom center of the page, there is a small copyright notice: '©PFC UAB - Samuel Jiménez 2010'.

Figura 5.10: Pàgina d'identificació

opcions que podem observar són:

- (A) Estat del servei: si està activat o parat
- (B) Número de dispositius detectats en les últimes 24 hores: independentment de si s'han arribat a realitzar la resta de processos o no, ens informa de quans dispositius mòbils amb el Bluetooth activat han estat dins del radi d'acció
- (C) Número de dispositius explorats en les últimes 24 hores: quantitat de mòbils que han contestat tenir el perfil del protocol OOP per tal de poder passar a la última fase. Aquesta quantitat, per regla, haurà de ser inferior al total.
- (D) Enviaments en les últimes 24 hores: quantitat de mòbils que han passat tot el procés i se'ls hi ha enviat el missatge de publicitat, sense considerar l'acceptació o error en aquesta fase.
- (E) Nom del fitxer a enviar: inclourà un enllaç al fitxer per poder veure quina publicitat s'està realitzant.

Com podem veure i per lògica, sempre es complirà que

$$A > B > C \quad (5.2)$$

degut al fet que un procés és dependent d'un altre i en tots existeixen filtres. De fet, és altament probable que el valor d' A sigui descarament més elevat que la resta ja que en general, qualsevol mòbil amb Bluetooth activat no tindrà problemes per respondre al paquet de *inquiry*. En canvi, la diferència entre B i C serà molt més reduïda ja que si es té el protocol activat, l'enviament és obligatori i, en principi, segur. Malgrat això, no seran equivalents degut a la gestió de recursos del sistema (pot ser que en un moment determinat hi hagin tants telèfons per rebre publicitat preparats que la meitat d'ells es quedin a la espera i en la següent ronda d'enviaments hagin aparegut nous dispositius cercats més recentment i, per tant, tinguin prioritat respecte els altres.

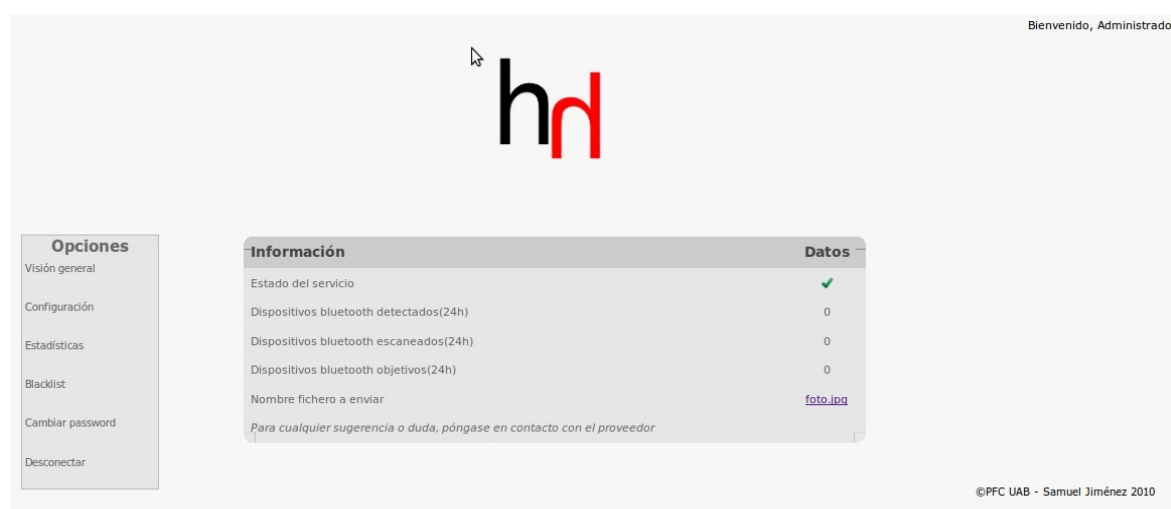


Figura 5.11: Pantalla inicial

L'objectiu d'aquest apartat és purament informatiu.

### 5.7.3 Configuració

Aquesta serà la pàgina en la que l'usuari interaccionarà més. L'administrador disposa de dos opcions a realitzar:

- Modificació de l'estat del servei: opció per encendre o bé parar el servei



- Substitució del fitxer de publicitat a ser enviat

En els dos casos, es crida a una nova pàgina que s'encarrega dels canvis i es retorna a la pàgina de configuració (figura 5.12, on es podran observar els canvis realitzats).

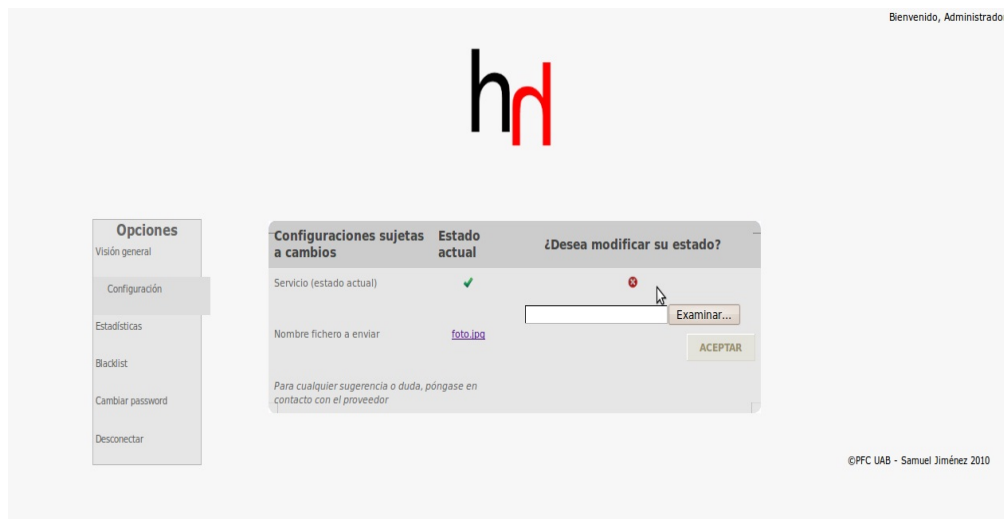


Figura 5.12: Pantalla de configuració

### 5.7.4 Estadístiques

En aquesta part, es realitza una representació visual de la informació inicial. Els usuaris solen preferir gràfics a dades numèriques ja que són més fàcils d'interpretar. En aquest cas, representem les dades comentades en el resum general.

### 5.7.5 Llista negra

Aquesta serà la pàgina de les prohibicions. L'administrador podrà donar d'alta o baixa a direccions MAC de telèfons coneguts. La utilitat d'aquesta funcionalitat és la d'evitar que a cada canvi de publicitat hi hagi un desgast en enviar aquesta als treballadors o persones molt properes a les que es vol evitar aquest tipus de contacte. Gràcies a aquest apartat, quan es fa una consulta per passar una MAC

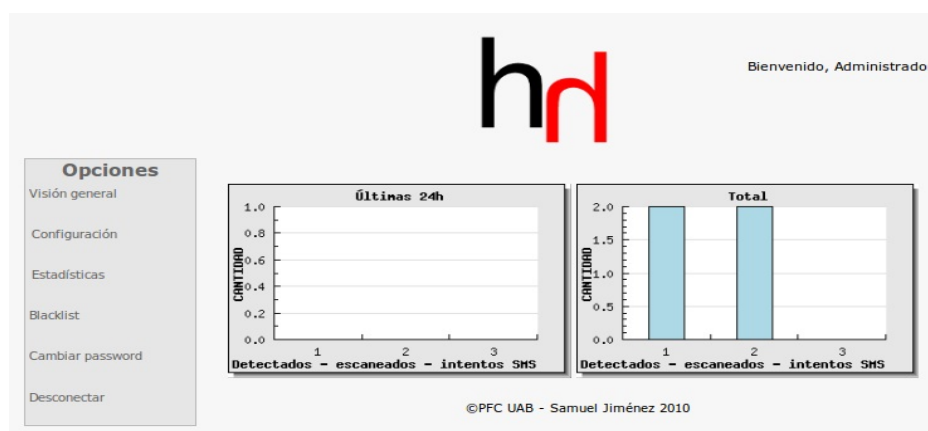


Figura 5.13: Pantalla d'estadístiques

per la fase 3 es comprova que no existeix a la llista negra. En cas d'existir, es desestima el recurs i es salta al següent objectiu. Sí passarà la fase 2 ja que una MAC no té perquè està sempre bloquejada i per tant, serà informació útil ja trobada en cas de desbloquejar-la en un futur.

### 5.7.6 Canviar paraula de pas

Per últim i per afegir un grau de seguretat en el control de la conta administradora, existeix la possibilitat de canviar la paraula de pas, altament recomanable de tant en quan per evitar problemes d'intrusions no consentides.

### 5.7.7 Rutines de control

Fins ara hem parlat de totes les accions importants i vistoses de cara a l'usuari però fan falta petites rutines que controlin el bon funcionament del sistema i que s'encarreguin de establir paràmetres globals. Així, hi ha un conjunt d'scripts que s'executen al inici del sistema (gràcies al *crontab* incorporat d'Ubuntu) per a dur a terme aquest objectiu:

- Control de l'estat del sistema: aquest script s'executarà cada minut per comprovar que l'usuari vol tenir el sistema engegat o parat. En cas de ser nec-

essari, aplicarà l'opció corresponent fent un

```
kill -9 <PID>
```

dels processos encesos o bé els executarà.

- Control del temps màxim en la fase 2: com hem comentat prèviament, amb el coneixement de què, a priori, en un interval de 3 segons un dispositiu ha d'haver contestat a la petició d'exploració, aquest script s'encarregarà de comprovar que si hi ha un procés de la fase 2 executant-se, s'elimini si fem un *timeout* del temps estipulat, per alliberar el recurs i poder saltar al següent objectiu.
- Càrrega del sistema: aquest script només és usat inicialment per carregar la base de dades.
- Distribució dels *dongles*: cada vegada que s'inicia l'activitat, es comproven els dispositius locals disponibles (per si algun d'ells ha fallat) i es reassignen amb una distribució adequada. Aquesta tasca permet que si teníem 2 dispositius destinats a la cerca i per motius desconeguts no estan disponibles en el moment d'iniciar el sistema, altres recursos seran assignats a la fase 1.

## 5.8 Problemes

En el desenvolupament del projecte hi han hagut diversos entrebancs de diferent vessant que han aportat problemes a un a l'hora de buscar una solució.

Primerament, existeix molta informació respecte la tecnologia Bluetooth però no sobre eines de desenvolupament. Amb prou feines hi ha empreses (sobretot a Espanya) que es dediquin a la publicitat per bluetooth de manera exhaustiva i, les que s'hi dediquen, no solen donar gaire informació per evitar competència. En aquest sentit, el fòrum del *elhacker.net* ha sigut molt útil ja que un dels desenvolupadors de l'empresa més avançada al respecte d'Espanya actuava com a moderador en l'apartat destinat a comunicacions mòbils i podia solventar dubtes conceptuals (sempre que no posés en perill el secret professional).

Més coses que han dificultat el projecte ha sigut la dificultat en aconseguir els enviaments paral·lels. Per exemple, per la fase 1 era impossible fer dos crides al sistema des de dos Shells obertes ja que una s'executava però l'altre indicava que tenia els recursos ocupats. Per tant, es va haver d'implementar un programa que creés dos processos per executar això, però el desenvolupament va ser fet a cegues degut a què no hi havia una assegurança de que allò arribés a funcionar.

Per altre banda, la mescla d'informació respecte el tema és molt gran. Especialment per la gran diversitat de versions. Com hem explicat en el capítol de l'estat de l'art, la pila que s'utilitza actualment es la de Bluez, però fa anys no era així (openObex era la més usada). Això comporta que no se sàpiga amb seguretat de quina pila estan parlant (i les dos tenen diferències substancials). A més a més, les funcionalitats implementades han variat i això fa que encara sigui més caòtic poder classificar quines estan vigents i quines no. A tall d'exemple, abans es requeria fer una associació per RFCOMM a l'hora d'explorar un dispositiu però això ja no és necessari.

Ha sigut especialment problemàtic l'enviament de fitxers. Primerament, era necessari evitar l'autenticació que implementa el Bluetooth el qual consisteix en establir un enllaç entre els dos dispositius introduint el mateix número PIN en els dos terminals. Degut a què aquest no es transmet per ràdio sinó que es calcula mitjançant diversos algorismes, es feia molt complicat *crackejar-ho*. Després d'investigar, es va trobar la solució del *uss-push* el qual és una minimització del protocol OOP en la què només es requereix confirmació per part de l'objectiu.

# Capítol 6

## Proves i resultats

Degut a què la complexitat d'anàlisis del sistema creix considerablement en augmentar el número de dispositius que interactuen amb el nostre servidor, s'ha decidit realitzar una simulació d'un entorn real però de manera controlada (amb 10 dispositius Bluetooth objectius). Així, serà més senzill extreure conclusions i que es pot extrapolar perfectament a un entorn més gran.

Després d'aplicar proves de temps sobre cada fase per detectar els temps requerits a cada etapa, s'han arribat als següents resultats:

El temps mig de detecció d'un dispositiu és **10,24 / (número *dongles* destinats a la cerca)** segons.

El temps mig entre la detecció d'un dispositiu i la seva exploració també és altament dependent dels recursos assignats, però posant una situació ideal en la qual l'entorn sigui estable<sup>1</sup> i el número de cerques és similar als recursos assignats (i per tant, no es faran llargues cues d'espera), és de **8,43** segons. Tot i això, aquesta és una dada molt poc fiable i faria falta una mostra gran de proves en molts entorns diferents ja que la naturalesa de l'entorn i les característiques dels telè-

---

<sup>1</sup>Hem de considerar que si els telèfons estan un interval de temps inferior a 2 segons dins del radi d'acció, els detectarem i mai arribarem a explorar-los. A més, el *time out* per saltar al següent objectiu és de 4 segons, fent que els temps mitjos de la resta de dispositius pendents augmenti considerablement.

fons fan ballar molt aquest resultat. A tall d'exemple, en un ambient control·lat i petit, la resposta més ràpida després de molts intents ha sigut de **3.3** segons i la més lenta, de **23** segons. Clar que això és totalment elàstic: en el cas de què no pareixin d'arribar nous mòbils, aquests últims **23** segons tendrien a infinit perquè la seva prioritat seria cada cop més baixa.

Podem conclure, per tant, que només podem establir un llinar inferior en el temps mínim d'exploració degut al fet de que el llinar superior no és directament calculable, a priori.

En la última fase, s'ha observat que el temps mig d'enviament d'un missatge (independentment de si és acceptat o no) és de **18.8** segons. De nou, totalment dependent del context i amb moltes variacions. Amb les mateixes raons d'abans, el llinar mínim s'ha fixat en **5.5** segons.

Si bé no s'ha extret dades empíriques en un entorn real, sí que s'ha llançat el sistema per comprovar el seu correcte funcionament i conclusions generals. S'ha pogut observar el flux de dades obtingut per ser analitzat.

## 6.1 Conclusions

A partir de les dades obtingudes, podem extreure varies conclusions:

El número de recursos assignats és la variable amb més pes en tot el sistema (a excepció de la fase 1 que la seva efectivitat no és dependent d'això). Per tant, a més dispositius, més **throughput** obtindrem. Només podem obtenir valors mínims, però mai valors màxims degut a la política de selecció d'objectius que minimitzar les probabilitats de congestió. El temps òptim teòric (assumint dos dispositius per a la cerca) és de:

$$(10, 24/2 + 3, 3 + 5, 5) = 13,92 \text{segons.} \quad (6.1)$$

I el temps mig és de:

$$(10, 24/2 + 8, 43 + 18, 8) = 32, 35 \text{ segons.} \quad (6.2)$$

Aquests resultats són molt importants de cara a l'empresa per a decidir el sector del mercat al que es vol dirigir: els consumidors han de tenir la possibilitat de contar amb tenir un client prop més de 30 segons aprop. En cas contrari, si l'ambient varia massa ràpid, no li interessarà i per tant, no s'aconseguirà vendre.

Tot i això, aquestes són les conclusions arribades amb un prototip que disposa de recursos limitats. La capacitat per afegir nous dispositius locals només depèn del número de ports USB disponibles en el servidor. Per tant, és de suposar que un model més potent reduiria notablement els temps, podent evitar la restricció de mercat esmentada.

Com a últim comentari, destacar que els mòbils no detectats (per les seves característiques com, per exemple, l'iphone, que té el Bluetooth restringit a només l'ús dels mans lliures) no han sigut considerats en aquest estudi ja que no hi ha manera de conèixer quins estan a l'entorn i quins no.

També podem calcular el throughput teòric que ha tingut el prototip. El màxim d'enviaments per minut ve donar per l'equació següent:

$$60/13,92 = 4,31 \text{ missatges/minut} - > \text{uns 4 missatges/minut} \quad (6.3)$$

En cas de calcular a partir d'un temps mig el resultat és diferent:

$$60/32,35 = 1,85 \text{ missatges/minut} - > \text{uns 2 missatges/minut} \quad (6.4)$$

Com podem veure, les diferències són significatives. Tot i això, no són determinants ja que el producte en venta tindrà més recursos i, per tant, tindrà un throughput més alt. Conseqüentment, el que sí podem afirmar és que el through-

put mínim serà de 2 missatges/minut o, el que és el mateix, 120 missatges per hora.

Finalment, s'ha arribat a la conclusió que el throughput és directament proporcional al número de mòbils existents<sup>2</sup> dins de la zona d'acció fins a un màxim, moment a partir del qual el throughput no pot augmentar i la corba es satura, tal com s'observa a la figura 6.1.

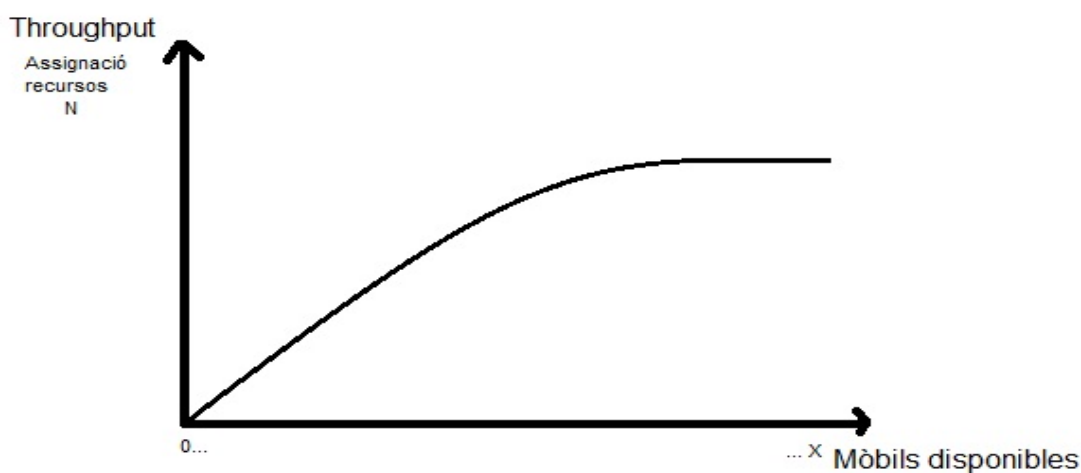


Figura 6.1: Gràfica punt de saturació del throughput

---

<sup>2</sup>Suposant que tots els telèfons tenen Bluetooth activat.



# Capítol 7

## Rendibilitat

S'ha fet un breu estudi per tal de determinar la rendibilitat del projecte.

### 7.1 Compra del sistema

Volem de descobrir a partir de quin punt obtenim beneficis mensualment. Per trobar el punt d'equilibri, s'aplica la fórmula següent:

$$Pe = CF/P - CV \quad (7.1)$$

On CF són els Costos Fixos, P el Preu de venda i CV els Costos variables del producte.

Pels costos fixos hem de contabilitzar el cost inicial de la programació pel primer període i els costos de actualització. El projecte ha estat realitzat en un període de 15 crèdits universitaris equivalent a 450h de treball durant el període de febrer a Juny (112,5h/mes). Aproximant un sou de 7.5€/h, veiem que els costos de personal sumen un total de 3.375€(843.75€/mes). A més, es suposa la continuació de l'activitat en el futur degut a la necessitat d'actualització permanent pel que es continuarà treballant el mateix temps establert, podent desenvolupar funcions de gestió comercial, distribució, implementació de millores...

El hardware actual té un cost de 565€(inclou processador i disc dur). El preu estimat de venda inicial és de 625€. Un cop ja tenim totes les variables podem

resoldre l'equació:

$$Pe = CF/P - CV = 843.75/(625 - 565) = 14,06. \quad (7.2)$$

S'han de vendre 15 unitats/mes per tal de que sigui una operació profitosa. Aquí s'haurien d'afegir els costos d'una llicència comercial per usar MySQL. Tot i així, degut a què es planteja una migració a PostgreSQL (sense costos de llicència) no s'ha tingut en compte. A més, el hardware espera una dràstica reducció de preu al escollir un altre element de manera que els costos variables baixarien a 300€, resultant en una reducció del preu a 399€ i un nou punt d'equilibri:

$$Pe = CF/P - CV = 843.75/(399 - 300) = 8,5. \quad (7.3)$$

Com veiem, el hardware és un element molt important en la rendibilitat del producte i per tant, ha de ser altament considerat. En aquest cas, s'ha reduït el preu en el 33% aproximadament i la necessitat de vendre s'ha reduït a només 9 unitats mensuals per obtenir profit.

## 7.2 Lloguer del sistema

A més de la venda del producte, s'ha pensat en posar-lo en lloguer de manera que en la celebració d'eleccions, cerimònies o concentracions podria ser utilitzat puntualment. El preu seria de 60€/dia i seria la opció més rentable per a l'empresa i econòmica per als clients. En aquest cas, s'hauria de fer un estudi individual segons les necessitats mensuals ja que la despesa inicial seria elevada per la compra d'unes quantes unitats del hardware però a partir d'aleshores i degut a que el preu d'actualització està inclòs dins les hores de treball del personal, només s'obtindria beneficis. Per a poder resoldre aquest problema s'ha usat el criteri del pay-back el qual és permet mesurar el temps que es triga a recuperar una inversió. No és totalment exacte ja que existeixen altres variables a considerar però ens pot donar una idea del temps de recuperació. Així, si partim de que això és un projecte d'inversió i que el cost del personal està directament associat a la venda de

nous productes i no al lloguer d'aquests, hem de resoldre la següent equació:

$$Pay-Back = -(10*565) + BeneficisMes1 + BeneficisMes2 + \dots + BeneficisMes12 \quad (7.4)$$

Suposant que  $Pay-Back > 0$  (per trobar el llindar dels beneficis) i que volem calcular un exercici (període de 12 mesos), hem d'obtenir una mitjana de 471€ per mes. Si el lloguer són a 60€/dia, s'hauran de llogar mínim 8 unitats/dia per mes per a que sigui una activitat profitosa al cap d'un any, o el que és el mateix, 95 unitats en un període de temps X per a que sigui rentable.

Si en canvi, contemplem el segon escenari amb un cost de 300€/unitat hardware, veiem que el resultat és molt diferent:

$$Pay-Back = -(10*300) + BeneficisMes1 + BeneficisMes2 + \dots + BeneficisMes12 \quad (7.5)$$

Aquí només seria necessari llogar 50 unitats/dia a l'any o bé 5 unitats/dia per mes.

## 7.3 Conclusions

Com podem observar, el projecte no necessita d'una alta capacitat comercial per part de l'empresa ja que a poques operacions que es duguin a terme és rentable. A més, amb el pas del temps els costos seran encara més baixos gràcies a la disminució d'hores necessàries en el sistema (encara que sempre existents) i per tant, el marge de beneficis serà superior a l'actual. No s'han inclòs els costos de les llicències ja que són prescindibles (en cas de migració a PostgreSQL i usar una alternativa lliure, encara que més difícil d'implementar, pels gràfics).



## Capítol 8

### Conclusions

Un cop analitzades les dades resultats de les proves en un entorn real i després d'haver dut a terme un profund treball d'investigació sobre la tecnologia Bluetooth, es confirma que pot ser una eina de màrqueting potent però limitada:

Potent gràcies al fet de que pot arribar a enviar molta publicitat a un cost molt baix a mig termini. Limitat, en el sentit que la tecnologia Bluetooth no és igual per a tots els mòbils i, per tant, es depèn d'aquests per a ser un producte efectiu.

A més a més, s'observa que els temps del inici al final del procés són molt variants. Però això no hauria d'importar sempre i quan el sistema estigui treballant, ja que no s'ha d'arribar a tots els dispositius que passin per la zona sinó al màxim número de receptors possible. La situació ideal seria que sempre s'estiguessin enviant missatges contínuament.

S'ha aconseguit tenir un ampli coneixement del Bluetooth, del qual podem dir que les seves opcions en el mercat són moltes i que amb el nou estàndard que està a punt de sortir, el qual serà molt més potent, les opcions es multiplicaran ja que solventarà algunes de les limitacions actuals.

Respecte la interfície gràfica, ha quedat demostrat que no és necessari una obra d'art per aconseguir quelcom funcional. Si bé és una mica menys impressionant, un disseny clar proporciona comoditat a l'usuari.

Degut als diferents processos que s'han de realitzar en el sistema, la base de dades és un element clau i la seva gestió, essencial. La integritat del sistema depèn

de la integritat d'aquesta.

Si bé en un inici sembla que el processador no és capaç de gestionar diferents dispositius a la vegada, una bona programació permet aconseguir-ho. El control entre processos és molt important per evitar problemes de coherència amb la base de dades, així com d'accès als recursos disponibles.

Ha sigut necessari programar en diferents llenguatges i estudiar diferents àrees de l'informàtica (base de dades, sistemes operatius, programació...) per, finalment, afirmar que les proves han sigut exitoses i que hem assolit els objectius del projecte al tenir un sistema capaç d'enviar a diferents dispositius mòbils en paral·lel i de realitzar totes les accions pertinents de forma autònoma sense la intervenció d'un administrador.

## 8.1 Millores

Després d'haver desenvolupat l'aplicació de publicitat per bluetooth han sorgit diferents propostes i millores al sistema actual que no han pogut ser implementades per la seva complexitat lligat amb el temps estipulat per al projecte. Tot i així, algunes d'elles afegirien un valor afegit gran al projecte i poden arribar a fer destacar el producte dins del mercat. No totes són de caire purament tècnic sinó que també s'ha contemplat la possibilitat de canviar elements del sistema per tal d'abaratir costos.

Les millores per optimitzar són les següents:

Fer el codi d'exploració dels telèfons en busca dels protocols disponibles i les seves característiques (*sdptool*) optimitzat. Això vol dir que es podrà evitar totes les consultes i passos intermedis innecessaris en aquesta fase degut a que només ens interessa saber el canal del protocol Obex push.

Fer el codi d'enviament de fitxers en lloc d'aprofitar una comanda predefinida. Actualment, el codi d'enviar un fitxer està basat en una comanda ja feta i molt limitada, la qual restringeix moltíssim les opcions del mercat (el seu abast) per esta dissenyada sense cap tipus de característica especial sinó únicament de manera estàndard per a la majoria de mòbils. Realitzant aquest procés de manera

personalitzada s'aconseguiria un enviament molt més flexible. Aquesta millora aconseguiria ampliar el públic objectiu.

No ha sigut possible la seva implementació degut a la alta complexitat que tindria fer un programa d'aquestes característiques ja que comportaria un estudi del funcionament de les principals marques de mòbil del mercat així com del funcionament intern d'aquest protocol (s'hauria de realitzar des de zero i actuant sobre *Bluez* directament).

Creació d'usuaris del sistema. Seria oportú ampliar la capacitat d'accés al sistema possibilitant la creació de nous usuaris els quals tinguessin certs permisos (com canviar la publicitat o encendre i parar el servei). Aquest nou concepte no va ser percebut en un primer moment de manera que no ha sigut implementat, però s'ha vist que seria positiu per a evitar deixar tots els permisos d'ús a una tercera persona a més de l'administrador.

Creació de LOGs del sistema. En conseqüència de la millora anterior, seria necessari un sistema de registres integrat a la interfície gràfica que permetés controlar els usos que li han donat al sistema els usuaris. Serviria com a control a la vegada que es podrien programar alarmes (enviar un correu quan passés l'esdeveniment X) per a saber què ha passat en tot moment.

## 8.2 Desenvolupament temporal

Com podem veure a la figura 8.1, la planificació temporal ha sigut modificada. Es pot observar que el desenvolupament de les diferents fases en la comunicació Bluetooth s'ha retrasat en 1-2 setmanes per cada fase, amb especial retràs en la última de totes degut a les complicacions que hi van haver. Conseqüentment, la redacció de la memòria i la preparació de l'exposició oral ha sigut fet més tard ja que no es podia avançar fins a acabar les parts anteriorment mencionades.

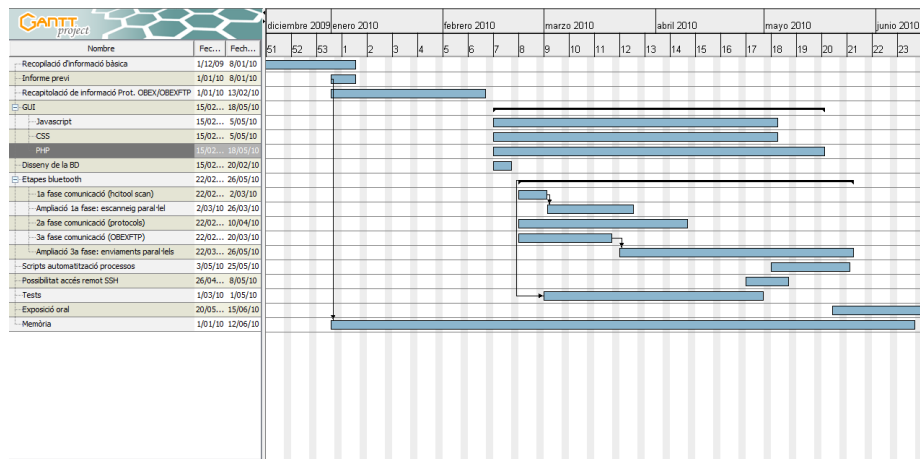


Figura 8.1: Desenvolupament temporal final



# Bibliografía

- [1] Official Linux Bluetooth protocol stack; *pila Bluetooth para linux*  
<<http://www.bluez.org/>>
- [2] Gadgets y tecnología; *qué es Bluetooth*  
<<http://tecnio.com/C2BFque-es-bluetooth/>>
- [3] English Wikipedia; *Bluetooth*  
<<http://en.wikipedia.org/wiki/Bluetooth/>>
- [4] English Wikipedia; *IOCTL*  
<<http://en.wikipedia.org/wiki/Ioctl/>>
- [5] Bluez Libraries; *Bluez source code*  
<[http://bluez-libs.sourceforge.net/documentation/2.15/hci\\_8c-source.html/](http://bluez-libs.sourceforge.net/documentation/2.15/hci_8c-source.html/)>
- [6] Wikipedia española; *Perfiles Bluetooth*  
<[http://es.wikipedia.org/wiki/Perfil\\_Bluetooth/](http://es.wikipedia.org/wiki/Perfil_Bluetooth/)>
- [7] Alberto Moreno; *Seguridad mobile*  
<<http://www.seguridadmobile.com/>>
- [8] Free Software Foundation; *Licencia GNU GPL*  
<<http://es.tldp.org/Otros/gples/gples.html/>>
- [9] Matrix, S.A.; *Componentes electrónicos*  
<<http://www.matrix.es/>>

- [10] PHP-Driven chart; *JpGraph*  
<<http://jpgraph.net/>>
- [11] Oracle; *Base de datos MySQL*  
<<http://www.mysql.com/>>
- [12] Apache Software Foundation; *Apache*  
<<http://ws.apache.org/>>

---

Firmat: Samuel Jiménez Díaz  
Bellaterra, Juny de 2010

## **Resum**

La tecnologia Bluetooth és una eina molt potent i cada cop més usada en el dia a dia. La possibilitat de portar-la cap a àmbits empresarials pocs explotats fins al moment pot ser una bona oportunitat de negoci. En aquest projecte, s'ha implementat un sistema d'enviament de publicitat a través de missatges a telèfons mòbils usant Bluetooth, el qual és capaç de gestionar-se de manera autònoma i només cal configurar-lo per una senzilla interfície gràfica accessible per xarxa. L'enviament concurrent ha sigut possible, fent que s'enviïn diversos missatges a la vegada.

## **Resumen**

La tecnología Bluetooth es una herramienta muy potente y cada vez más usada en el día a día. La posibilidad de llevar-la a ámbitos empresariales poco explotados hasta el momento puede ser una buena oportunidad de negocio. En este proyecto, se ha implementado un sistema de envío de publicidad a través de mensajes a teléfonos móviles usando Bluetooth, el cual es capaz de gestionarse a sí mismo de forma autónoma y sólo necesita configurarse a través de una sencilla interfaz gráfica accesible por red. El envío concurrente de múltiples mensajes ha sido posible, haciendo llegar la publicidad a varios teléfonos a la vez.

## **Abstract**

Bluetooth technology is a very powerful tool that has been increasingly used in the normal life basis. There's a lot of potential to include it in business areas unexploited so far. In this project, we have implemented a publicity mailing system by mobile texts using Bluetooth. It can completely auto manage itself. You only need to set it up with a network user-friendly graphic interface. It has also proven successful when sending multiple texts at once to several mobile terminals.